Variable transformation to a 2×2 Domain Space for Edge Matching Puzzles

Aspinall, Thomas; Gepp, Adrian; Harris, Geoffrey; Vanstone, Bruce J

Published in:
Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices - 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Proceedings

DOI:
10.1007/978-3-030-55789-8_19

Variable Transformation to a 2×2 domain space for Edge Matching Puzzles

Thomas Aspinall[1 [0000-0002-6968-1989]], Adrian Gepp[1 [0000-0003-1666-5501]], Geoff Harris[1 [0000-0003-4284-8619]], and Bruce James Vanstone[1 [0000-0002-3977-2468]]

Bond Business School, Bond University, Gold Coast, Australia.
taspinal@bond.edu.au, adgepp@bond.edu.au, gharris@bond.edu.au, bvanston@bond.edu.au

Abstract. The Eternity II (E2) challenge is a well-known instance of the set of Edge Matching Puzzles (EMP), which are examples of combinatorial problem spaces of the worst-case complexity. Transformation of the domain space to consider pieces at the 2×2 level increases the total number of elements but is shown to result in orders of magnitude smaller search spaces. While the original domain space has uniform cardinality, the transformed space exhibits statistically exploitable features. Two heuristics are proposed and compared to both the original search space and the raw transformed search space. The efficacy of the two heuristics is empirically demonstrated. An explanation of how the mapping results in an overall decrease in the number of nodes in the solution search space of the transformed problem is outlined.

Keywords: Worst-case Complexity, Transformed Search Space, Edge Matching Puzzle, EMP, Eternity II (E2) challenge

1 Introduction

A new methodology for pre-processing EMPs, in which the variables and domain elements are transformed into a different EMP problem, is presented. The use of heuristics to exploit the structure of the transformed domains are investigated.

EMPs belong to the NP-complete (NP-C) problem set. Depending on the objective function chosen, solution optimization is an NP-hard problem [1]. This work focuses upon two-set Framed Generic Edge Matching Puzzles (GEMP-F) [2] that adhere to the structural characteristics of the Eternity II problem (E2-style). The E2-Style GEMP-F has an $n \times n$ board with n^2 slots and four-edged pieces and possess a distinguishing border pattern and two sets of patterns for edge and inner matches [2]. Each tile must be placed such that pairwise adjacent tiles have matching patterns. E2-style GEMP-F are designed to eliminate statistical information inherent within the characteristics of both the pieces and distribution of edge patterns. This feature is designed to minimize the ability of Solution Algorithms to reorder domain variables via some form of metric function in order to improve search space traversal efficiency.

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

This paper presents a transformation of the variables such that the resulting domains do exhibit statistical weaknesses not apparent in the original E2-style GEMP-F problem sets. This involves considering all valid 2×2 tiling combinations that can be generated from four single tiles within a problem instance. The transformed EMP problems have several properties that make each instance easier to solve. First, transforming these problem instances results in a collapse of the depth of the search space tree by a factor of four. Second, unlike the original variables, the domains of the transformed variables have statistically exploitable features allowing heuristic search tree trimming thus enabling directed searching towards solutions. The results of this study demonstrate that performing the domain transformation and reordering tiles according to the presented heuristics does reduce the median number of nodes traversed to approach a global solution. This approach effectively solves the smallest sized puzzles, with the impact of the transformation becoming less pronounced for larger puzzle sizes as the total number of variables within the domain space increases.

After examining the E2-style GEMP-F literature (Section 2), the impacts of variable transformation on the EMP problem space and its generation of statistically exploitable features are presented (Section 3). Two efficient search space traversal heuristics are outlined (Section 4). The results of applying these heuristics to problem instances are presented and discussed (Section 5). Future research avenues are outlined in Section 6.

2 Literature Review

E2-style problems were developed following the claimant of the £1 million cash prize offered for the solution of the first Eternity puzzle [3]. The winners of the original Eternity prize were contracted to collaboratively develop the Eternity II puzzle and were likely given the task of eliminating the combinatorial flaws used to solve the original puzzle. The result was an GEMP-F, of size $n = 16$, possessing structural characteristics that cannot be as readily used to reduce the problem search space. Harris et al. [4] describes a process for generating E2-style GEMP-F instances. The 3-partition problem has been proven to be NP-Hard [5], with E2-style puzzles the worst-case instances of NP-hard problems [2].

Strategies for solving E2-style GEMP-F have used algorithms belonging to methodologies such as Evolutionary Algorithms [6], Boolean satisfiability [2, 7, 8] and constraint satisfaction [9, 10]. These algorithms employed heuristics and meta-heuristics, e.g. tabu search, look-ahead, back jumping and arc consistency, to reduce total search space size and optimize the scores.

Extending the Eternity II challenge, the 3rd International Conference on Metaheuristics and Nature Inspired Computing (META'10, 2010), provided four online downloadable E2-style GEMP-F instances of size $n = \{10, 12, 14, 16\}$ [11]. The winning approach of the META'10 challenge, which coincidentally did not fully solve any of the problem instances, was the application of a two-phase guide-and-observe hyper-heuristic solution algorithm with a score of 461/480 on the size $n = 16$ problem instance [12]. The heuristics utilized involved swap and

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

rotate moves at both the single, double and triple tile level. The work of Wauters et al. [13] considered the number of inner matches for all 4×4 regions in the puzzle, an extension of the 3×3 objective that provided promising results. These experiments showed that by first optimizing an objective besides the raw score of the puzzle and switching to this objective, a higher average score is reached [13]. The work of Salassa et al. [14] applied a hybrid local search approach through both mixed-integer linear programming and Max-Clique formulation, building complete solutions by constructing optimal sub-regions across the problem space. The application of both greedy and backtracking constructive heuristics resulted in maximum scores of 459/480 for the commercial E2 instance and the META'10 size $n = 16$ problem instance [14].

Recently it was shown that the computational overheads involved in heuristics that reduce search space size are greater than those for brute force implementations imposed upon a static variable instantiation scheme [4]. This is due to E2-style EMP's design having no statistically exploitable weaknesses. The Zero-Look Ahead (ZLA) algorithm was shown to outperform all other Solution Algorithms (in 40 of the 48 benchmarks) and was up to 3 orders of magnitude faster than previous published solvers [4].

Many variable transformation techniques have been used while the area of NP-Complete problems has been studied, e.g. variable transformation for Latin Hyper-cubes [15] and in published attempts at solving the real-world Car Sequencing Challenge, with instances provided by Renault [16]. Given the widespread opinion that $P \neq NP$ (e.g. [17]) each transformation is a problem specific creation. To date, there have been no published variable domain transformations for the E2-style EMP. This paper develops such a transformation specifically for the E2 problem and which can be applied to all other EMPs.

3 Variable Transformation

The domain elements of the variables for E2-style GEMP-F problems are single tiles, each being square, non-symmetrically patterned and unique. Each tile rotation (C4 geometric symmetry) represents a valid placement into the variables of the problem space. The elements of an EMP are each piece in each possible rotation, providing the total variable space. Satisfaction of the constraints of each placement determines if the final configuration (all variables instantiated as shown in Fig. 1) represents a valid solution or not to the problem. Consider all 2×2 tiling combinations to be generated from 4 single tiles (Fig. 1). A 2×2 piece is the valid aggregation of four individual pieces all with matching inner edges. The resultant 2×2 combination is an individual piece (domain element) of a higher order.

The characteristics of 2×2 pieces used when further solving EMP are its outer edge patterns and the four individual pieces making up the transformed variable (Fig. 1). If all possible 2×2 combinations are generated then the original single tile $n \times n$ problem can be transformed into an $\frac{n}{2} \times \frac{n}{2}$ problem (assuming n is even – a requirement for an EMP problem to belong to the E2 class).

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

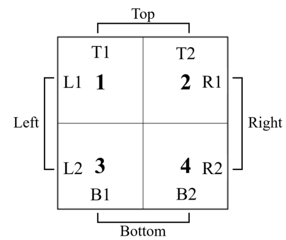

Fig. 1. 2×2 tiling generated from four single tiles.

The number of variables has been reduced from n^2 to $\frac{n^2}{4}$. As the combinatorial problem space grows with n, this reduction in the number of levels in the search space tree is associated with a combinatorial decrease in the number of nodes in the transformed search space.

Note that transforming the variables of an EMP to the 2×2 domain space results in more variables available for assignment than there are positions on the board. The transformed problem instance thus involves the assignment of a subset of these variables to the domains of a 2×2 piece board of size $= \frac{n}{2}$. A constraint upon the variables of an EMP is that each single piece is used once only. As 2×2 pieces comprise four individual pieces, when a 2×2 piece is placed onto the board, all other 2×2 pieces that feature any identical individual pieces are removed from the variable domain space. This 2×2 domain transformation produces an important effect upon piece types, i.e. corner, edge and inner pieces. Piece types at the 2×2 level feature a combination of one, two or all three piece-types at the individual level (see Fig. 2).

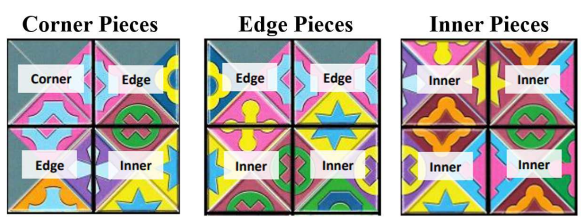

Fig. 2. The Three 2×2 Piece Types.

The distinction between solving for the frame and inner sections of the board is blended at the 2×2 level as assigning frame pieces onto the board reduces available inner pieces and vice versa. This characteristic of 2×2 pieces results in a collapse of the branching factor of the expanded search space as 2×2 pieces are assigned. This leads to faster backtracking of Solution Algorithms, increasing the speed of search space traversal.

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

Although the domain transformation results in a collapse of the number of levels of the search space, the number of branches from each node in the transformed 2×2 tiling space increases in conjunction with the increased number of elements. Sadly, no known analytic method can determine, a priori, if the trade-off between: (a) the decrease in tree depth and (b) the increase in the average branching factor at each node in the transformed tree, results in a significant reduction in search space size. Instead empirical examination of randomly generated E2-style GEMP-F is required.

Pre-processing pieces by performing a domain transformation is classified as pre-processing as the propagation of variables within CSP Solution Algorithms need be applied only once per problem instance. Processing time for the pre-processing of pieces before they are utilized in a Solution Algorithm is, therefore, not considered in evaluating the efficiency of Solution Algorithms.

Table 1. GEMP-F 2×2 Frequency of Piece Types and Pattern IDs.

Puzzle Size	Corner Pieces	Edge Pieces	Inner Pieces	Total Patterns
4	180	0	0	32
6	727	5,138	6,866	73
8	2,138	26,570	77,108	113
META'10-10	2,280	58,388	342,046	180
META'10-12	3,536	136,832	1,033,686	240
META'10-14	3,512	165,592	1,888,540	376
META'10-16	5,180	290,864	4,079,776	459
Eternity II-16	5,824	292,012	4,059,952	459

The number of elements of 2×2-EMP elements demonstrates neither a combinatorial nor exponential increase in size with increasing n (Table 1), leading to the possibility in net increase in Solution Algorithm performance as a result of the corresponding combinatorial decrease in the number of nodes in the search space. Efficient traversal and indexing of these elements will be necessary to effectively utilize 2×2 elements within a Solution Algorithm. For solving 2×2-EMP's to be more efficient than for the original (untransformed) EMP, the benefit of an increase in efficiency of search space traversal will have to outweigh the cost of the increase in total variable domain sizes.

3.1 Effect Upon Edge Pattern Distribution

One of the properties of E2-style GEMP-F is a uniform distribution of edge patterns. That is, there is no statistical information about the distribution that could be used to determine the order in which to instantiate variables (assuming a SAT/CSP methodology). The pattern distribution of the original Eternity II puzzle is shown in Fig. 3.

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

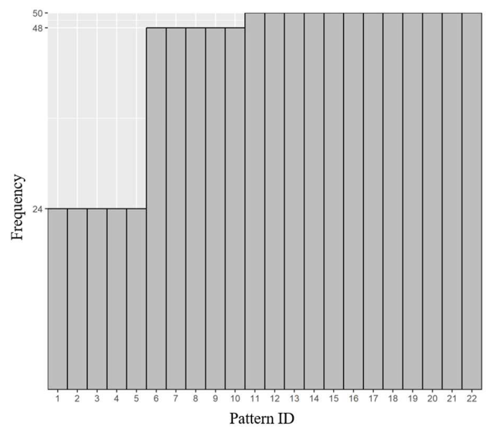

Fig. 3. The Eternity II Puzzle Pattern ID Distribution.

The consequence of this uniformity of pattern distribution is to maximise entropy thereby making variable selection heuristics and guided search techniques ineffective at both reducing and identifying solution rich regions of the search space. In terms of statistical frequency, there is no reason to target pieces of one pattern over another. If the pattern distribution was not uniform, domain minimizing heuristics can be used to select pieces that are the most constrained (have the least number of patterns of the puzzle). These are the pieces most likely to collapse the search space and instigate backtracking. This is very similar in methodology to the first Eternity puzzle solver [3]. Nonetheless, the pattern distribution for the transformed variable space exhibits statistical features that can be exploited. Fig. 4 shows the distribution of the Eternity II puzzle in its 2×2 space.

The most important result of the transformation to the 2×2 phase space is the effect this transformation has on the pattern frequency distribution of the 2×2 elements. The difficulty of the commercial E2 instance was optimized through careful assignment of both the number of patterns and the distribution they possess. The patterns were not however optimized in any way at the 2×2 level as is dramatically shown in Fig. 4.

The pre-processing of E2-Style GEMP-F by generating 2×2 elements introduces statistical information within the pattern distributions and variable domains. Thus, by definition, pre-processing elements has transformed the problem instance into one that is no longer an E2-style problem. The non-uniform distribution of the edge patterns in the transformed puzzle enables the use of heuristics and CSP techniques, such as arc consistency and dynamic ordering of variable instantiation, to reduce the search space compared to a pure brute force approach. This systemic process of traversing the search space, as opposed to randomly instantiating elements and traversing through root nodes, results in an increase in Solution Algorithm performance.

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

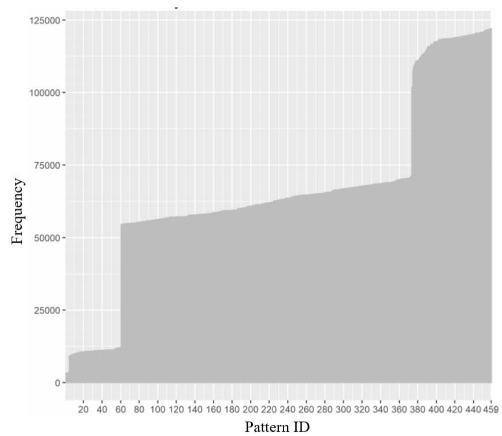

Fig. 4. The Eternity II Puzzle 2×2 Pattern ID Distribution.

At the 2×2 level the presence of more elements than variables on the board might allow statistical information to be used to distinguish between solution and non-solution pieces. If this is possible, then there may be ways to reorder the variables of the elements to search elements that feature in a solution rich search space. Furthermore, as 2×2 pieces (elements) are placed onto the board, the domains and subsequent pattern distribution of remaining domain elements change, since piece placement in the 2×2 problem results in immediate domain size reduction across all related non-instantiated domains. This suggests two domain variable ordering heuristics, which are now proposed and discussed.

4 Domain Variable Ordering Heuristics

Two variable selection heuristics are proposed to filter the problem space to search in regions that contain a statistically higher likelihood of containing a global solution. These heuristics do so by increasing the rate at which solution 2×2 pieces that are a subset of a global solution are analysed by a 2×2 Solution Algorithm. A domain characteristic of 2×2-EMP is that it possesses more 2×2 piece elements than there are positions upon the board for variable assignment. Thus a subset of these 2×2 elements belong in a global solution, and, therefore, belong to a solution rich region of the search space. If the benefit of a more efficient search space traversal outweighs the computational cost of possessing all 2×2 pieces within the domain space, then the net Solution Algorithm performance will increase. This is as a result of pre-processing elements to the 2×2 space and running variable selection heuristics upon them.

The two variable selection heuristics are inspired by Bayesian statistical theory and attempt to determine the probabilistic likelihood that a 2×2 element matches a given position. The goal of reordering the variable elements by these

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

metrics is to rank them by this probabilistic likelihood, to be used in the filtering of domains. Call the two heuristics H1 and H2, with H2 being an iterative improvement upon H1. The heuristics construct metrics equal to the product of the domain sizes of each border edge, i.e. the number of 2×2 pieces that can form a valid placement. Fig. 5 displays the edge matches per 2×2 piece type.

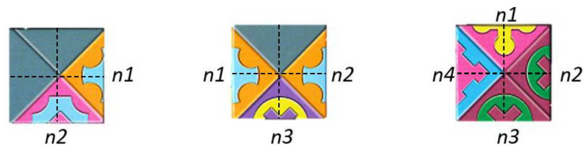

Fig. 5. Domain Ordering Metrics per 2×2 Piece Type.

Heuristic 1 (H1): the product of the domain sizes of each border edge.

$$\prod n_i \tag{1}$$

Then index the 2×2 pieces according to their metric to enable the piece with the highest metric value to be chosen first during the traversal of the search tree. Eliminate any 2×2 piece for which the metric is 0. The higher the value of H1, the more likely a 2×2 element can be placed into a partial solution, as the metric has calculated more adjacent matches to this element than others. Fig. 6 provides an example of the application of H1 with valid matches.

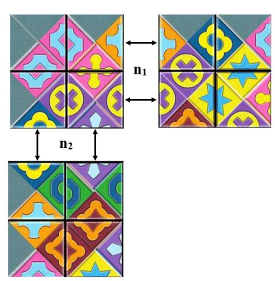

Fig. 6. Heuristic 1 applied to a 2×2 Corner Piece.

Heuristic 2 (H2): the product of the domain sizes of each border edge such that pieces with common single tiles are not counted twice.

$$\prod (n_i - x_i) \tag{2}$$

Index the 2×2 pieces by metric to enable the piece with the highest metric value to be chosen first during the traversal of the search tree. Eliminate any

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

2×2 piece for which the metric is 0. The additional condition of Heuristic 2 is that the elements that make up matching 2×2 pieces must be unique. Fig. 7 provides an example of valid and invalid pairings of 2×2 pieces. Although the edge patterns for n_1 form a valid match of these 2×2 pieces, the individual elements that make up these 2×2 pieces are not unique, resulting in an invalid match of these 2×2 pieces. The edge patterns for n_2, however, form a valid match with no duplicate elements, thus forming a valid match of these 2×2 pieces. The additional constraint provided by H2 results in a product function that is always lower than H1 at the cost of more computationally intensive pre-processing.

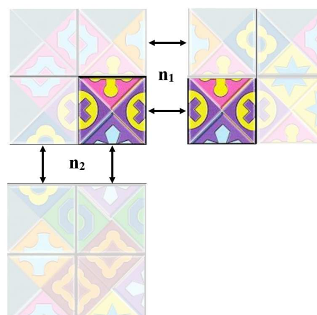

Fig. 7. Example of valid (n_2) and invalid (n_1) pairing.

5 Results and Discussion

The solution generation algorithm applied at both the single and 2×2 tile level was a brute force approach featuring no forward checking, back jumping or k-arc consistency, as first proposed in the work of Harris et al. [4]. This algorithm has been empirically determined to outperform all previous results that utilized the aforementioned domain trimming tactics [4]; it was argued that this is a consequence of E2-style GEMP-F possessing zero statistically exploitable features. For this initial work on the transformation of the variable space, it was determined to utilize this particular Solution Algorithm for both the untransformed and transformed problem spaces.

Solution Algorithms for the single tile and 2×2 tile approaches were implemented in Pascal on a current generation Intel$^{\text{TM}}$$Core^{TM}$i9-8950HK CPU @ 2.90GHz. The heuristics were applied to border 2×2 pieces within this work and were encoded separately to the Solution Algorithm, computed as the 2×2 input pieces were generated from the corresponding single piece element file. Consequently, no runtime search time was required for the various heuristics; it was entirely absorbed in the input file generation phase. These implementations of the Solution Algorithm output the number of nodes traversed before a

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

global solution was generated for each problem instance presented to it. This enabled a direct comparison between the median number of nodes (as suggested by Ansótegui [7]) traversed to find a solution as a function of E2 puzzle size n.

Sets of 20 randomly generated single tile E2-style GEMP-F instances were created of puzzle sizes $n = \{4, 6, 8\}$ with differing border and inner colour combinations (see Table 2). Table 2 presents the results of running the Solution Algorithm at both the single and 2×2 tile level, with two variable domain heuristics applied at the 2×2 level.

Table 2. Comparison of the Effect of Nodes Traversed by Puzzle Size, Domain Transformation and Domain Ordering Metrics.

Puzzle Size	4	4	6	6	6	8	8
Border : Inner	3:3	3:4	3:5	3:6	3:7	3:6	3:7
Single Tiles	88	63	305,428	179,379	25,421	14,012,064	542,359,516
2x2 Tiles	14	6	103,648	94,499	8,800	11,151,356	399,094,862
Heuristic 1	10	5	91,730	101,713	4,789	4,967,412	433,661,359
Heuristic 2	7	4	83,689	96,052	5,052	6,309,757	393,869,489

The results above clearly demonstrate the efficacy of the heuristics, particularly when applied to lower puzzle sizes. Indeed, we can see that the size $n = 4$ puzzles are for all intents and purposes a solved problem: the reordering of the tiles according to either heuristic when approached as a search space spanning problem solves the puzzle. In all of the other puzzles we can see that the inclusion of additional constraints to the heuristics leads to a reduction in the total number of nodes traversed per solution.

The general trend of the above results indicates that the impact of the transformation was less dramatic for larger puzzle sizes. In this work, the heuristics were only applied to the border pieces and although the border pieces increase linearly with puzzle size n, the number of inner pieces increases at least quadratically as a function of n. It is thus not too surprising to note that for this initial study, applying the heuristics to solely the border pieces has its greatest impact when there are no inner pieces (size $n = 4$) and the least impact for the largest puzzle instances (size $n = 8$).

6 Conclusion and Future Work

This paper has presented a transformation of the variable and domain space for E2-style GEMP-F which is shown to exhibit statistical structure which should be able to be used to reduce the size of the search space. Two heuristics are proposed to exploit this observed structure, albeit of just the border pieces, and an empirical study demonstrates their effectiveness. It is noted that the impact of the heuristics is smaller for larger n, a consequence of the border

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

pieces increasing linearly in n whilst the inner pieces grow quadratically. Given the results above, and taking into account the factorial increase in nodes per solution as a function of n, it is not surprising that in the literature to date there has been no report of solving a single $n = 10$ instance of the E2-style GEMP-F. Given this is the first attempt to transform the E2 problem into a problem with statistically exploitable features, these initial results appear very promising. It is fair to assume that similar heuristics, developed specifically for the inner pieces, would reduce the nodes traversed in especially the $n = 8$ puzzles, bringing the overall effectiveness more into line with the observed results of the $n = 4$ and $n = 6$ puzzles.

The potential research tree is vast, especially as pre-processing elements to a 2×2 phase space on Eternity II has not been explored previously. This includes extensions of the present empirical analysis; development of a 2×2 Solution Algorithm; development of further 2×2 variable selection metrics, especially for inner pieces, and the construction of analytical proofs of the empirically observed characteristics of EMP in this research. Calculating multiple global solutions to an EMP instance may provide associated insights into why specific 2×2 elements are filtered to the top of the reordered variables. If it were observed that the metric filtered solution pieces across multiple solutions that the EMP possesses, it would be further evidence of changes to solution density of the search space at the upper percentiles of the reordered variables.

The value of pre-processing 2×2 elements of an EMP as well as the scaling ability of the Solution Algorithm could be empirically verified through the development of a 2×2 Solution Algorithm. This Solution Algorithm could be used to analyse net change in search space efficiency and solving time as a result of transforming the domains of EMP's and the reordering according to suggested variable selection heuristics. This would require an efficient indexation of the large numbers of 2×2 variables as well as update the variable space to eliminate invalid elements as pieces featuring identical individual pieces were placed into a solution. This would also provide evidence as to whether or not the transformed problem space features puzzle hardness equal to an E2-style GEMP-F.

The creation of new domain reordering heuristics is suggested to further determine the likelihood that an element belongs to a certain position on the board. Performing Monte-Carlo simulations as the 2×2 Solution Algorithm traverses the search space of an EMP could be used in an attempt to extract statistical information inherent within the domain variables of elements. This would determine the probabilistic likelihood that an element is a subset of a global solution in each position on the board of the EMP. The implementation of a 2×2 Solution Algorithm, using Monte-Carlo simulations to generate this new variable selection metric, is the next natural step recommended to continue this research.

Finally, it is proposed that the use of a transformation to 3×3 pieces be investigated and applied to $n = 6$ puzzles. This may very well, in conjunction with the heuristics proposed above, reduce the $n = 6$ puzzle into a solved problem, much as has been demonstrated here for the $n = 4$ puzzle sizes.

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19

References

1. Demaine, E.D., Demaine, M.L.: Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. Graphs and Combinatorics **23** (2007) 195–208
2. Ansótegui, C., Bonet, M.L., Levy, J., Manya, F.: Measuring the hardness of sat instances. In: Twenty-Third AAAI Conference on Artificial Intelligence. (2008) 222–228
3. Selby, A., Riordan, O.: Eternity - description of method. `http://www.archduke.org/eternity/method/desc.html` (2000)
4. Harris, G., Vanstone, B., Gepp, A. Lecture Notes in Computer Science (LNCS). In: Automatically Generating and Solving Eternity II Style Puzzles. Springer, Germany (5 2018) 626–632
5. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences). W. H. Freeman (1979)
6. Muñoz, J., Gutierrez, G., Sanchis, A.: Evolutionary genetic algorithms in a constraint satisfaction problem: Puzzle Eternity II. In Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M., eds.: Bio-Inspired Systems: Computational and Ambient Intelligence, Berlin, Heidelberg, Springer Berlin Heidelberg (2009) 720–727
7. Ansótegui, C., Béjar, R., Fernández, C., Mateu, C.: On the hardness of solving edge matching puzzles as SAT or CSP problems. Constraints (2013) 1–31
8. Heule, M.J.H.: Solving edge-matching problems with satisfiability solvers. In Kullmann, O., ed.: International Conference on Theory and Applications of Satisfiability Testing. (2009) 69–82
9. Schaus, P., Deville, Y.: Hybridization of CP and VLNS for eternity II. Journées Francophones de Programmation par Contraintes (JFPC'08) (2008)
10. Bourreau, E., Benoist, T.: Fast global filtering for Eternity II. Constraint Programming Letters (CPL) **3** (2008) 036–049
11. : Meta 2010 problem instances. `http://web.ntnu.edu.tw/ tcchiang/publications/META2010-Instances.rar` (2010)
12. Vancroonenburg, W., Wauters, T., Vanden Berghe, G.: A two phase hyper-heuristic approach for solving the Eternity-II puzzle. In: Proceedings of the 3rd International Conference on Metaheuristics and Nature Inspired Computing (META). Volume 10. (2010)
13. Wauters, T., Vancroonenburg, W., Berghe, G.V.: A guide-and-observe hyper-heuristic approach to the Eternity II puzzle. Journal of Mathematical Modelling and Algorithms **11**(3) (2012) 217–233
14. Salassa, F., Vancroonenburg, W., Wauters, T., Croce, F.D., Berghe, G.V.: MILP and max-clique based heuristics for the eternity II puzzle. CoRR **abs/1709.00252** (2017)
15. Liefvendahl, M., Stocki, R.: A study on algorithms for optimization of latin hypercubes. Journal of Statistical Planning and Inference **136**(9) (2006) 3231 – 3247
16. Solnon, C., Cung, V.D., Nguyen, A., Artigues, C.: The car sequencing problem: overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. European Journal of Operational Research **191**(3) (2008) 912–927
17. Aaronson, S.: Guest column: NP-complete problems and physical reality. ACM Sigact News **36**(1) (2005) 30–52

The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-030-55789-8_19