**ANALYSIS OF AN ETERNITY II PUZZLE SOLUTION THROUGH THE IMPLEMENTATION OF GENETIC ALGORTIHMS**

*M. J. Nitzken*

CECS545: Artificial Intelligence, University of Louisville

Thursday, December 2, 2010

#### Abstract

The Eternity II is an edge matching puzzle invented in 2007. The puzzle functions by connecting pieces divided into sectors together with like sectors to generate a final coherent mapping. The complexity of the actual puzzle is immense and with 256 pieces making up the original puzzle it is estimated that brute force techniques would require more than 2 x 1047 steps to locate a solution. The puzzle is classified as a type of NP-complete decision problem and thus adding pieces exponentially increases the overall complexity. This paper thus proposes a comparison utilizing an elitist genetic algorithm to attempt a solution on an Eternity II type puzzle. The algorithm is compared utilizing a cost maximization approach versus a wisdom of crowds approach for solving the puzzle. Various parameters and settings as well as different cost functions are examined in the pursuit of a solution to an Eternity II style puzzle.

**Index Terms—** Eternity II, board game solution, genetic algorithm, mutation, wisdom of crowds

**1. Introduction**

The Eternity II puzzle is a 256 piece edge matching puzzle board game that is licensed and sold by the TOMY Corporation. The puzzle was invented by Christopher Monckton and was released on July 29, 2007 [[1](#_ENREF_1)]. The puzzle has garnered international attention thanks to the $2 million prize that has been offered for finding the correct and complete solution to the puzzle. According to the official websites the puzzle has still not been correctly solved and no submitted solution has ever been found to be completely free of errors. The Eternity II puzzle is also the successor to the original Eternity puzzle.



Fig.1. The Eternity II puzzle showing a partial solution.

The puzzle mechanics are similar to a standard edge matching puzzle game. The game consists of 256 square shaped pieces that are each divided by two diagonal lines at ±45º. This effectively divides a piece into 4 triangles with each one facing inward and the bases facing the outside of the square piece. A piece can have any combination of 22 different colored tiles on it. The color combination and order is not limited meaning that a piece may have the same color in all 4 positions or a random dispersion of colored pieces. An edge piece will have a black triangle on a side which connects to the outer edge. A corner piece therefore has two such black edges indicating it to be a corner. Pieces are placed such that the outside of the board contains a black border from the edge pieces and with all interior pieces connecting to a matching color piece. The puzzle was specifically designed to make brute force methods through the use of a computer nearly impossible to accomplish. The actual puzzle is also complicated by the presence of a "starter" piece which must be placed at the center of the playing board and which all other pieces must surround.

The Eternity II game is actually a single instance of a much larger NP-Complete problem. The original form of the Eternity II is the game of Tetravex which is a modified version of the MacMahon squares puzzle [[2](#_ENREF_2), [3](#_ENREF_3)]. The MacMahon squares puzzle follows a similar but greatly simplified gameplay strategy. This comparison is important in determining the NP-Completeness of the Eternity II problem.

**1.1. Genetic Algorithms**

A solution to a form of Eternity II (EII) puzzle is proposed through the use of genetic algorithms [[4](#_ENREF_4), [5](#_ENREF_5)]. A genetic algorithm is a search heuristic that is comparable to the process of evolution found naturally in our world [[6](#_ENREF_6), [7](#_ENREF_7)]. The system works by evolving potential solutions toward an ultimate goal. The evolution takes place through the processes of inheritance, mutation, selection and crossover [[7](#_ENREF_7), [8](#_ENREF_8)]. Due to the complexity of the EII puzzle and the challenges associated in designing a solver this paper will focus on an asexual variant of a genetic algorithm. This decision has been made due to the complexity in exchanging individual pieces between separate boards as without a large amount of added complexity the chance for game piece duplication and deletion in the populations becomes very great if not inevitable in future populations making a solution impossible to find.

**1.2. Wisdom of Crowds**

The Wisdom of Crowds approach to solution solving was proposed by James Surowiecki and published in 2004 [[9](#_ENREF_9), [10](#_ENREF_10)]. The book presents numerous studies indicating that while a single individual in a population may never score perfectly the combined group when pooled in consensus performs better than any one individual. This proposal can be applied to a genetic algorithm in computer science. By taking the combined intelligence of the population of solutions it is proposed that a better solution than any single individual could be found to the desired search task. This paper will further discuss the design and implementation of a Wisdom of Crowds technique combined with a genetic algorithm as an approach for solving the EII puzzle.

**1.3. Additional Approaches**

Several other approaches have been attempted to solve the EII problem [[11](#_ENREF_11), [12](#_ENREF_12)]. Because the original game was designed to be difficult for a computer to solve, researchers have sought to decrease the overall complexity of the search space so as to find a solution in a reasonable amount of time.

One method tried is a Constraint Programming (CP) model [[13](#_ENREF_13), [14](#_ENREF_14)]. In this model the program focuses to design a more efficient method of brute forcing the total solution space. The final technique involved a combination of local searching techniques to find the optimum piece placements. The solution was capable of scoring 458/480 pieces correctly but a perfect solution was never reached. While this is a potentially viable technique for solving the EII it still focuses on the usage of a traditional style search combined with a large amount of brute force calculation.

Another attempt focused on a comparison between an Artificial immune evolutionary algorithm and a Multiobjective evolutionary algorithm [[15](#_ENREF_15)]. These are variations of genetic algorithms that focus on the use of a population to attempt to solve the overall problem. The approaches discussed in this paper will also follow a genetic based approach although they will use different styles of genetic algorithms. I will focus primarily on a more complex fitness calculation and the usage of an elitist and Wisdom of Crowds genetic algorithm to attempt to formulate a solution to this problem.

**2. Methods**

In this paper a novel technique for solving a EII puzzle is proposed. The technique focuses on the usage of a genetic algorithm implemented on an EII object to attempt to find the best possible solution by maximizing the search space.

**2.1. Problem Formulation**

To perform an analysis on an EII puzzle object it was first necessary to construct a method of analyzing the perceived puzzle. The solution attempt was made using a MATLAB implementation. Not having access to an original EII puzzle I designed a simplified method of constructing and testing EII boards. This was done by generating a blank EII board and filling the edges with a value of 0 representing a black cell. While the original EII featured over 22 different total colors I simplified my board to include only 4 different colored pieces. While the framework is easily expandable to use the full number of colors this greatly simplified the computational load on the servers and calculation time required to test a board. Pieces were generated by randomly selecting a color and then adding the random color to the corresponding piece. The board was moved through constructing several pieces at a time until the entire board was filled.

While this board was greatly simplified from the original EII for calculation purposes it is important to note that this still acts as an NP-Complete problem and can easily become exponentially more challenging. While a computer may be able to easily solve a 2x2 puzzle increasing the board dimensions to 10x10 or NxN results in a solution that can be verified in polynomial time.



Fig.2. The simplest form of the MacMahon squares game, a type of Tetravex puzzle.

The reduction can be proven to be NP-Complete through the nature of the simplification and a comparison to the formal definition of NP-Completeness. By definition a problem can be determined to be NP-Complete by being shown to be more complex than a known NP-Complete problem. The Eternity II puzzle is a variation of the Tetravex puzzle which is constructed from the MacMahon squares game as seen in Fig.2. [[13](#_ENREF_13)]. The combination of these squares constructs a Voronoi diagram. The nature of this problem is in itself NP-Complete as discussed by Takenagaa and Ansotegui[[2](#_ENREF_2), [16](#_ENREF_16)]. The simplest form of the MacMahon squares puzzle, an NP-Complete problem, involves the use of 3 square colors arranged in a way similar to that of the Eternity II puzzle across an NxN grid (originally a 4x6 grid). This game has been proven to be NP-Complete. The proposed problem, while simpler than the Eternity II, uses more colors than the original MacMahon square problem and is therefore more complicated to calculate. It is also therefore an instance of the MacMahon square and Tetravex puzzle. Additionally the board is capable of equally growing to a size of NxN. Therefore while the proposed puzzle is a simplified variation of the Eternity II puzzle it is still more complex than the MacMahon squares puzzle which is NP-Complete and therefore the simplification retains the NP-Complete nature of its parent problem.

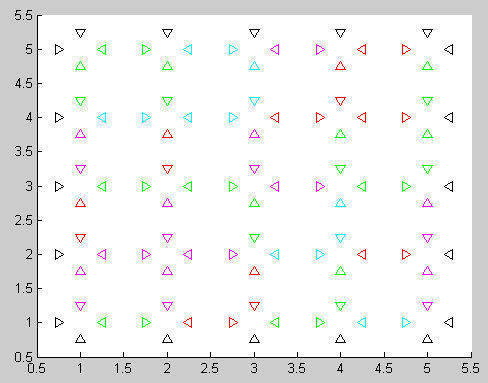


Fig.3. A sample EII board generated showing 4 different interior colors and black boundaries. Each piece consists of 4 triangle pointing in toward one another.

To effectively store the constructed board special objects were designed to efficiently manipulate the data. The board was represented by one 3-dimensional matrix of size NxNx4 that was used to hold the individual color values of a piece in the [N E S W] configuration at rotation position 0 and a second 2-dimensional matrix of size NxN used to hold a integer value corresponding to the rotation position of a piece. A piece would be rotated by this integer value the number of positions clockwise for actual piece positioning comparison with a value 0 being no rotation and a value of 3 rotating the piece three times so that the northernmost position faced the western edge. This allowed the values in the board to remain unchanged and static despite the location of a piece. In this way when attempting to construct a Wisdom of Crowds representation unique pieces could always be established commonly and quickly between different board arrangements.

In the construction of test populations a double-blind population generation was used. An initial perfect board was constructed to ensure that the puzzle was actually solvable. These pieces were then randomized several thousand times using location swaps and rotations of individual pieces to an intermediary board. This intermediary board was then mutated several hundred times during the construction of each board that would exist in the initial population. This ensured that the starting population was grossly different from the original board but that the search solution did in fact have an ultimate result and that the puzzle was not unsolvable.

**2.2. Genetic Algorithm Construction**

The EII storage object was uniquely designed to interact with the accompanying genetic algorithms. The genetic algorithm (GA) approach used was constructed in a similar manner to those used to solve a Traveling Salesperson (TSP) problem. While roughly based off of the TSP variation the large amount of changes to the search space and type of data being analyzed forced the algorithm to ultimately become very different from previous versions.

The GA was constructed in an elitist fashion that was ultimately driven by a Survival of the Fittest (SotF) based approach. In this form only the top percentage of a population is allowed to reproduce in future populations. The worst solutions from a population are therefore excluded and have no impact on the children. The methodology in selecting such an approach is that it is hoped that by excluding worse populations the algorithm will eventually pursue only the most profitable paths.

*2.2.1 Genetic Algorithm Fitness*

The fitness of a path was determined by the overall calculated cost of an individual solution. The decision was made to attempt to maximize the cost by rewarding the algorithm for performing correctly and penalizing it for poor performance. To different cost calculations were constructed in an attempt to maximize the performance of the algorithm. The first was a simple matching cost analysis while the second provided an in-depth scoring mechanic with a more complex technique for assigning scores to pieces.

In the simple cost analysis only pieces that matched on the board were counted as a score of 1. Pieces that did not match were given a score of 0. Initially I felt that due to the speed this technique would provide an adequate means for attempting a solution. Unfortunately this technique also rewarded the occurrence of matching outer edges together in the center of the playing board and I therefore created a more complex cost analysis to compare against.

The complex cost analysis functioned by assigning different values between edge, corner and inner pieces and penalizing the board for placing pieces in incorrect locations. This was done by comparing multiple edges of pieces to surrounding pieces and their locations. If a colored edge was matched correctly than the score of the overall board was increased by 2 for each correct matched piece. If a piece placed a black edge correctly on an outer edge it received a score increase of 1 and therefore a correctly positioned corner received a score increase of 2. To further encourage the algorithm to move edge pieces to the edges if a piece had correct color edges and was in the interior of the board but had any black edges it received a score change of 0 regardless of the number of correct edges. Finally matching two black edges together in the interior resulted in a score change of -1 as this could never happen in a final solution and was greatly discouraged.

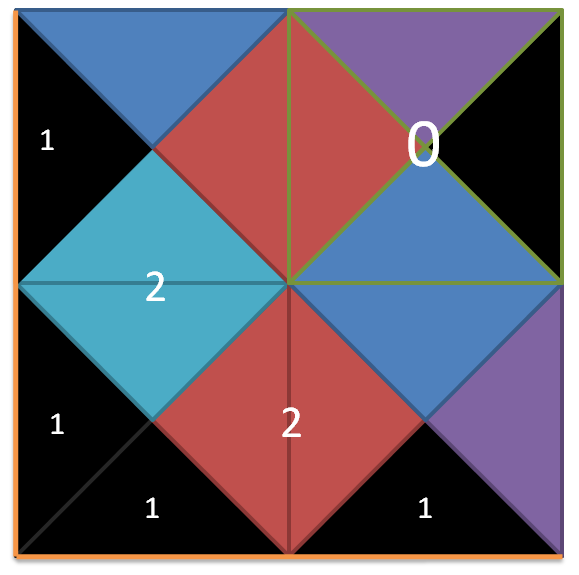
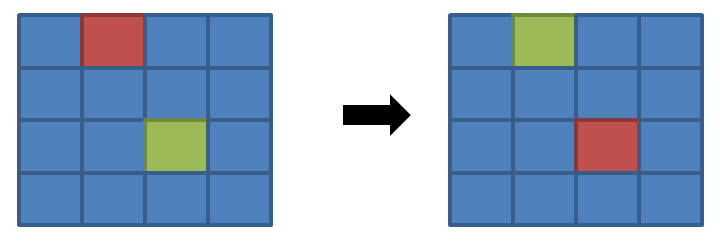


Fig. 4. Illustration of complex scoring. The board outer boundary is along the orange border. The correct boundary pieces are awarded 1 point and matching colors for pieces in a correct possible location are awarded 2. The piece highlighted in light green is a border pixel incorrectly placed in the interior and thus it is awarded 0 regardless of its 2 matching sides.

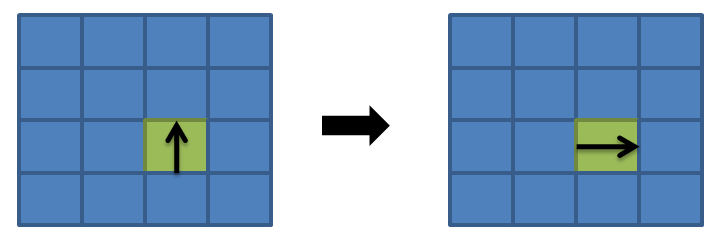
Following a mutation phase of each algorithm each board in the population could then be scored and the boards could be compared.

*2.2.1 Genetic Algorithm Mutations*

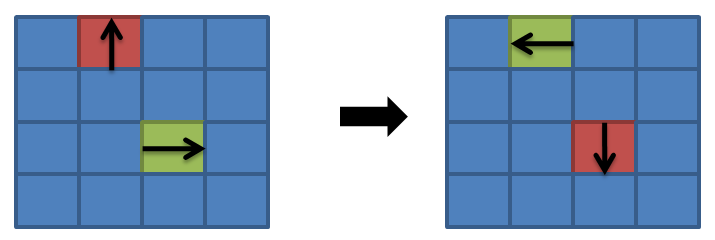
To perform a genetic algorithm it was necessary to construct a series of potential mutations that could be undergone. Three mutations were constructed as in Fig.5.; a tile exchange mutation, a tile twist mutation, and a tile exchange and twist mutation. The tile exchange mutation operator functioned by selecting 2 tiles and exchanging them within a board. The twist mutation operator functioned by selecting a tile and twisting it to a new position on the board. The twist direction change was determined at random being a twist of 1-3 positions. Lastly a combination of the previous two methods could occur in that two tiles were both repositioned and twisted simultaneously.

****

**(a)**

****

**(b)**

****

**(c)**

Fig.5. Piece mutations: (a) a tile exchange mutation, (b) a tile twist mutation, and (c) a tile exchange and twist mutation.

*2.2.1 Survival of the Fittest Metric and Wisdom of Crowds*

Two different metrics were examined for the selection of the best board from the population. These were referred to as the Survival of the Fittest (SotF) selection method and the Wisdom of Crowds (WoC) method. The SotF method worked by calculating the cost for each board in a population. This was done using one of the two previous scoring parameters. The results were then order-ranked and the best board was declared the winner.

The Wisdom of Crowds proved to be more of a challenge as it involved combining the knowledge of numerous boards to create a collectively selected board. The greatest challenge in this stemmed from reconstructing a board that featured only the original set of tiles and ensuring that no tiles were added or removed during the WoC selection process. To accomplish this a framework was constructed to map the boards against one another. Initially the first board was selected. Using this board the number of unique pieces and the number of occurrences of each piece was calculated. This served as the unique standard that would ensure that the final WoC board was a real board and contained no corrupted tiles.

Each unique piece was given an identification index and a 3-dimensional matrix of NxNxINDEX was constructed to store the occurrences of a specific piece on the board. Additionally rotation was important and therefore a second 4-dimensional matrix of NxNxINDEXx4 was constructed to track the most common rotations of a piece in a given location. Using these matrices a form of greedy algorithm was applied to solve for the WoC final board. This functioned by order-ranking the occurrence of pieces in a specific location of the board and finding the most likely to occur piece and its corresponding rotation that had not been previously used. Each time a piece was placed on the board it was removed from the unique identification pool to ensure that it could not be reused. The resulting board was the combination of the most popular piece locations and rotations among the entire population.

I also decided to implemented a final decision form of the Wisdom of Crowds to see how it compared to the other two techniques. In this method the genetic algorithm was simply run normally using the standard approach with the SotF metric. It ran till convergence and simply aggregated the occurrence data across the lifetime of the algorithm. By aggregating this data it tracked how many times each piece occurred in a given location with a specific rotation. This provided a much larger data pool for the wisdom of crowds to examining without drastically increasing the processor load. Tracking the algorithm for several hundred generations allowed for several thousand boards to be examined. It was assumed that as the population became more accurate it should have more desirable boards occur more often and therefore these boards and piece locations would eventually become the most prominent occurrences on the board. The original WoC algorithm was simply split into two halves and interjected at different cycles in the genetic algorithm to perform this change. Its comparative results will be discussed in the results section.

**3. Experimental Results**

The data was tested in a variety of techniques. As previously mentioned the simplistic scoring method did not generate adequate results to warrant cumulative tests therefore all tests were done using the complex scoring system. Two different boards were examined, a 5x5 and a 7x7 using both algorithms. In the 5x5 board several parameters were examined to find the best performance parameters. These were then examined on the 7x7 board. A population size of 500 was used in test cases. Both techniques were compared against a standard of pure randomness. In a pure random test 30,000 boards were randomly constructed from the original tiles. The highest score from this test was max random score. All statistic scores can be found in Table.1.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  | | **Board Score** | | | **Iterations** | | |
| **Type** | **Top %** | | **Mutate %** | **Attempt %** | | Min | Max | Average | Min | Max | Average |
| **Board Size = 5x5** | | | | | | | | | | | |
| **RANDOM** | |  |  |  | - | | 38 | - | - | - | - |
| **GA** | | 30 | 40 | 70 | 55 | | 57 | 56.3 | 180 | 359 | 261.3 |
| **WoC** | | 30 | 40 | 70 | 34 | | 41 | 36.3 | 132 | 178 | 154.7 |
| **GA** | | 25 | 50 | 40 | 57 | | 60 | 58.4 | 118 | 323 | 211.6 |
| **WoC** | | 25 | 50 | 40 | 36 | | 41 | 38.4 | 119 | 356 | 202.6 |
| **GA** | | 25 | 40 | 40 | 63 | | 64 | 63.7 | 247 | 378 | 305 |
| **WoC** | | 25 | 40 | 40 | 37 | | 44 | 41 | 231 | 475 | 326 |
| **WoC Aggr** | | 25 | 40 | 40 | 7 | | 27 | 18.3 | - | - | - |
| **Board Size = 7x7** | | | | | | | | | | | |
| **RANDOM** | |  |  |  | - | | 100 | - | - | - | - |
| **GA** | | 25 | 40 | 40 | 141 | | 144 | 142.3 | 199 | 256 | 231.3 |
| **WoC** | | 25 | 40 | 40 | 97 | | 102 | 99.7 | 135 | 301 | 213.3 |
| **WoC Aggr** | | 25 | 40 | 40 | 20 | | 65 | 43.2 | - | - | - |

Table.1. Results of performance tests

The Top %, Mutate %, and Attempt % are used to define the top percentage of the population to use in the offspring, the percentage the mutation will occur and the maximum percentage of the board that can undergo mutation.

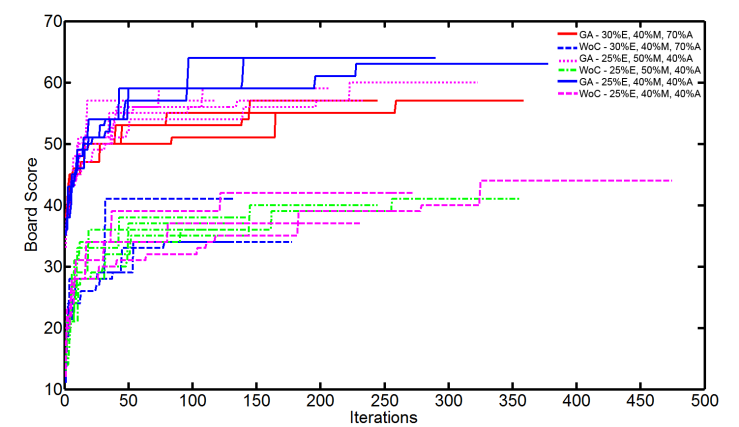


Fig.6. Board Score vs. Iteration (5x5 Board)

From each test run plots of the Board Score vs. Iteration were generated. It is easily noticeable that all of the standard SotF genetic algorithms are grouped at the top while the WoC genetic algorithms are clustered at the bottom for the 5x5 board as seen in Fig.6. Similar results were observed in the 7x7 board although the percentage difference in score was noticeably higher as seen in Fig.7. In many cases the WoC algorithm performed at or only barely above the level of the random board generations. It is interesting that the WoC initially performs at a level far below the random iterations and that through improvements reaches the same level as a random board. This phenomenon will be further discussed and reasons will be examined for this occurence.

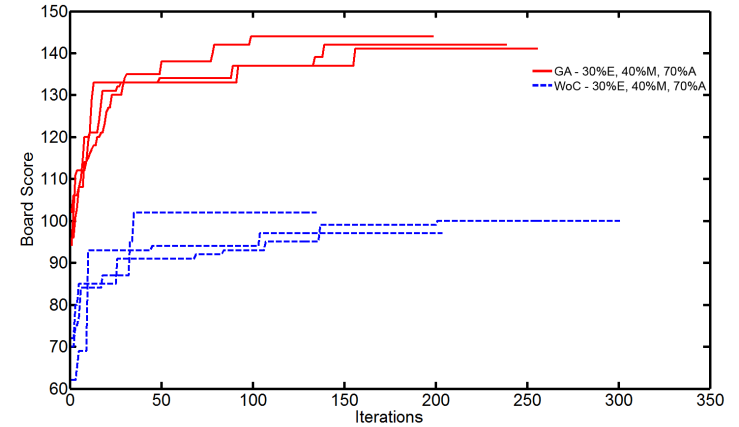


Fig.7. Board Score vs. Iteration (7x7 Board)

Overall, the results were not entirely as expected. In many cases a Wisdom of Crowds algorithm is believed to perform better than a standard genetic algorithm although this was clearly not the case for this form of problem.

The areas that the algorithm performed better on appear to be directly related to the scoring algorithm used. An example board can be seen in Fig.8. The board showed largely positive results around the edges and while it did not always orient the corners in the correct direction it generally would get the pieces in a corner location. The edge pieces also generally ended up near or on an edge.

I found the results of the aggregated WoC board applied at the end of the algorithm to be disappointing as well. I had believed that by driving the algorithm for an extended period it would ultimately place the correct pieces in location more often than not. I feel that the poor results of this may have been impacted by the starting boards. If an incorrect piece were to fall in a location too many times before it was changed it could negatively impact multiple pieces.

The impact on multiple pieces comes from the fact that even if the piece was moved to the correct position late, both the initial incorrect location and the new correct location would recieve the wrong values as they would have too high an occurence of the incorrect value to count correctly. This problem could be further compounded if one were to consider that say 5 pieces should be interchanged in linear fashion in this manner. In retrospect this technique would be too heavily weighted when using a convergent form of the algorithm. A better way to examine this style of occurence would have been to run the algorithm for 5-6x the iterations required for convergence. While this would have taken significantly more time it could yield a much higher accuracy for this method of analysis.

I found that regardless of the board created blue and green tiles generally tended to be matched up at a rate of twice or more than the red and magenta tiles. This occurred in nearly all tests regardless of the random board generated. I believe a possible cause of this occurrence may lie in the way that the pieces are structured, although this is largely due to random change. It appears that the blue and green based pieces tend to show up more commonly as duplicate colors on individual pieces than other colors. As such a piece containing a purple or red was also likely to contain a black, blue or green and only one red or purple piece whereas many generated pieces with a blue or green are likely to have a second blue or green piece adjacent. As mentioned this is likely a result of random luck and the reasoning is not quantifiable.

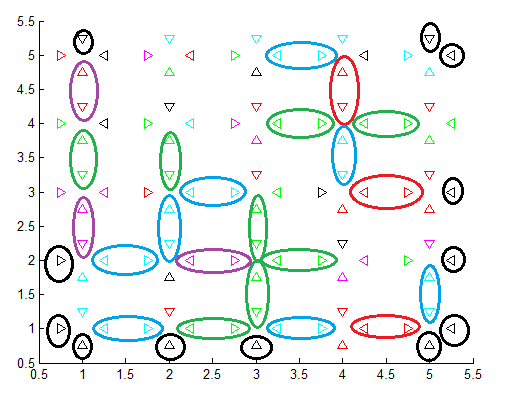


Fig.8. An end solution with correctly connected tile sides circled in corresponding colors.

**4. Conclusion**

While none of the solutions were able to perfectly solve the EII puzzle the SotF algorithm showed the most promise of the examined techniques. It may seem odd that given the way Wisdom of Crowds operates it would perform worse than a standard algorithm although I feel it is its innate property that gives it a significant disadvantage in this particular usage scenario.

A Wisdom of Crowds approach works by pooling together different boards to find a common unique board. In the tests due to the underwhelming speed of the algorithm the population was restricted to 500 boards. While this may be a low population size I feel that even a population of 10,000 or more may demonstrate poor overall results. The total number of potential boards must be carefully considered. In this type of NP-Complete problem there are billions or more possible board combinations. Using populations of a small size where each individual piece can have hundreds of thousands of different values the likelihood of any piece and exact orientation occurring more than a handful of times even in a moderately large population is slim. To adequately get a large enough occurrence rate very large populations would be required for the Wisdom of Crowds approach to show significant results. Unfortunately I did not have access to computers capable of performing such a large calculation load and was therefore unable to test this hypothesis. In my case and the population size it was highly likely that the Wisdom of Crowds was essentially constructing almost completely random boards and that the benefits of the genetic algorithm and population progression were not reflected.

The SotF genetic algorithm approach fared much better. The WoC explanation would actually explain the positive results from the SotF algorithm. These boards would directly benefit from the genetic mutation and still be impacted even in a small population. An individual board was capable of representing the entire populations positive results.

In consideration of the findings I feel that while the algorithms performed opposite of the desired effect that they did perform in an expected way considering the parameters used in the test cases. In future tests I would like to examine the impact of using a much larger population pool to attempt to solve the puzzle. Additionally if I could acquire an actual EII puzzle I think it would be interesting to see how the algorithm would perform on the actual board game puzzle.

**11. ReferenceS**

1. TOMY. *Eternity II Official Website*. 2010; Available from: http://us.eternityii.com/.

2. Takenagaa, Y. and T. Walshb, *TETRAVEX is NP-complete.* Information Processing Letters, 2006. **99**: p. 171-174.

3. Yasuhiko Takenaga, A., *Information Processing Letters 99 (2006) 171–174 TETRAVEX is NP-complete.* 2006.

4. Ansótegui, C., et al. *Edge matching puzzles as hard SAT/CSP benchmarks*. 2008: Springer.

5. Demaine, E. and M. Demaine, *Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity.* Graphs and Combinatorics, 2007. **23**: p. 195-208.

6. Haupt, R., S. Haupt, and J. Wiley, *Practical genetic algorithms*. 1998: Wiley Online Library.

7. Goldberg, D., *Genetic algorithms in search, optimization, and machine learning*. 1989: Addison-wesley.

8. Russell, S. and P. Norvig, *Artificial intelligence: a modern approach*. 2009: Prentice hall.

9. Surowiecki, J. and M. Silverman, *The wisdom of crowds.* American Journal of Physics, 2007. **75**: p. 190.

10. Kostakos, V. *Is the Crowd's Wisdom Biased? A Quantitative Analysis of Three Online Communities*. in *Computational Science and Engineering, 2009. CSE '09. International Conference on*. 2009.

11. Lee, W.K. *The six-piece Chinese jigsaw puzzle: an inspiration to structuring communication*. in *Professional Communication Conference, 2002. IPCC 2002. Proceedings. IEEE International*. 2002.

12. Toyama, F., et al. *Assembly of puzzles using a genetic algorithm*. in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. 2002.

13. Schaus, P. and Y. Deville, *Hybridization of CP and VLNS for Eternity II.* Journées Francophones de Programmation par Contraintes (JFPC’08), 2008.

14. Makridis, M. and N. Papamarkos, *A New Technique for Solving Puzzles.* Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2010. **40**(3): p. 789-797.

15. Muoz, J., G. Gutierrez, and A. Sanchis. *Evolutionary techniques in a constraint satisfaction problem: Puzzle Eternity II*. in *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*. 2009.

16. Ansótegui, C., et al. *How Hard is a Commercial Puzzle: the Eternity II Challenge*. 2008: IOS Press.