

Tantrix: A Minute to Learn, 100 (Genetic Algorithm) Generations to Master.

Keith L. Downing
Department of Computer Science (IDI)
The Norwegian University of Science and Technology
7020 Trondheim, Norway
keithd@idi.ntnu.no

July 22, 2005

1 Abstract

The game of Tantrix™ provides a challenging, mathematical and graphic domain for evolutionary computation. The simple task of forming long loops of colored arcs quickly becomes a search nightmare for humans and computers alike as the number of game pieces scales linearly. This paper introduces Tantrix-GA, a genetic algorithm that solves several types and sizes of Tantrix puzzles but still falls well short of (at least a few) human Tantrix experts. By introducing this problem to evolutionary computation researchers, we hope to motivate an evolutionary attack on the holy-grail Tantrix puzzles, one of which has yet to be solved by any intelligence, real or artificial.

Keywords: Tantrix, Genetic Algorithms, Indirect-Encoded Genomes

2 Introduction

In 1987, Mike Mcmanaway, a New Zealand backgammon champion and puzzle-shop owner, invented *The Mind Game*, a two-person board game involving two-colored hexagonal tiles. The game enjoyed regional popularity and by 1991 had evolved into *Tantrix*, a versatile set of 4-color tiles supporting both a 2-4 player game and host of individual puzzles. A set of 5 *Super* puzzles was added in 1994, the Tantrix internet site (www.tantrix.com) appeared in 1996, and the first world championships were held in 1998. Tantrix now appeals to a broad international audience, with many of the best players coming from Hungary, Israel and New Zealand.

Basic Artificial Intelligence (AI) has been on the scene since 1999, when the first automated players of multi-person Tantrix were released. Today, the Tantrix web site includes a "robot" competition in addition to the highly competitive human tournament.

The purpose of this paper is twofold: 1) to introduce Tantrix as an interesting and enjoyable domain for evolutionary computation (EC) teaching and research, and, 2) to present a first approach to solving Tantrix puzzles with a genetic algorithm.

The wide range of Tantrix puzzles can be co-opted to provide everything from a) simple and nicely graphical examples of indirectly-coded GAs, to b) challenging EC homework and project assignments, to c) extremely difficult search problems in highly-deceptive landscapes, to d) currently unsolved holy-grail puzzles that would probably tax even the largest EC-dedicated Beowulf.

This paper introduces Tantrix-GA, a genetic algorithm [7, 5] for solving Tantrix puzzles. We have used it to solve all 5 of the Tantrix *Rainbow* Puzzles, the 3- (trivial) to 30-block (very difficult) *Discovery* Puzzles, and the 5 Super Puzzles. The *Unsolvable* puzzles are currently well-beyond its reach. So, in addition to gaining basic insights into the application of GA to Tantrix, the reader may become inspired to join the hunt for solutions to the most perplexing Tantrix brain-twisters.

3 Tantrix Basics

The Tantrix *Game Pack* consists of 56 hexagonal tiles, each containing 3 arcs of different colors. Each of a tile's 6 edges is intersected by one of the 3 arcs. The tiles are numbered from 1 to 56, and the 4 groups of tiles 1-14, 15-28, 29-42, and 43-56 have a special significance with respect to the coloring scheme. All told, there are 4 possible arc colors in the 56-tile set: red, green, blue and yellow. However, each of the 4 groups of 14 employs a different set of 3 colors: 1) red, yellow, blue; 2) red, yellow, green; 3) red, green, blue; and 4) blue, green, yellow, respectively. To view the complete Tantrix tile set, see www.tantrix.com.

Orthogonal to this color scheme is a second 5-color labelling of the tile numbers. Each tile's number is written in one of these 5 colors on the back of the tile. There is no simple explanation for the assignment of these 5 colors, but they partition the 56 tiles into 5 sets: 1) Green - 10 tiles, 2) Yellow - 12 tiles, 3) White - 9 tiles, 4) Blue - 10 tiles, 5) Red - 15 tiles. These groups are the basis for the Rainbow puzzles explained below. In general, any reference to, say, the blue tiles refers to these 10

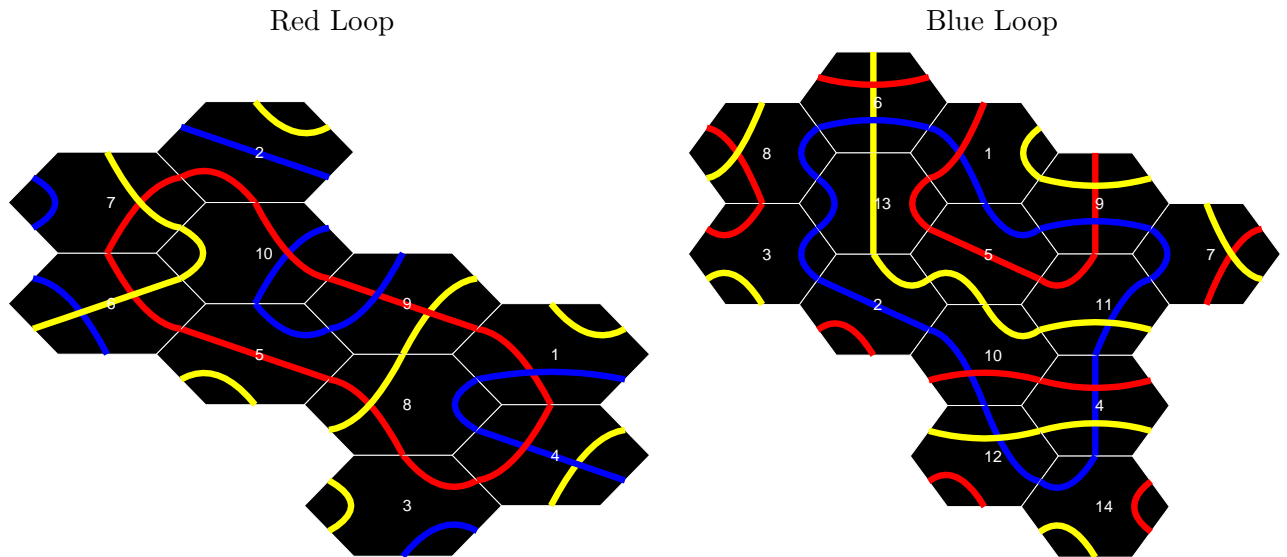


Figure 1: Solutions to the 10- and 14-tile Discovery puzzles.

tiles and not to all tiles containing a blue arc.

In general, the goal of all Tantrix games and puzzles is to form a single two-dimensional cluster of tiles (called the *tantrix*) containing long lines or loops of the arc colors, while abiding by *The Golden Rule*: the shared edge between any two tiles must have the same arc color on both sides. For example, the tantrix on the left of Figure 1 shows a red loop involving tiles 1-10. This constitutes a solution to the 10-puzzle from the Discovery set. Similarly, the rightmost tantrix solves the 14-puzzle, using a different color, blue.

Along with The Golden Rule, a second constraint is important in puzzle solving: the tantrix cannot contain holes, i.e., open cells completely enclosed within the tantrix. Figure 2 shows a 10-tile tantrix that contains a hole and thus is not a valid solution.

The Golden Rule and the hole restriction are the only hard constraints in Tantrix puzzle solving. A puzzle then consists of a given set of tiles, e.g. all 10 green tiles, and a (very general) description of the desired pattern, e.g., *a loop containing all 10 green arcs*. In no case is the exact shape of the complete curve or cycle given in the problem description, but in a few cases, the form of the tantrix is specified as a pyramid. Otherwise, the cluster's shape is also unconstrained, as long as it has no holes.

In the multi-person Tantrix game, each player has a different color and tries to form long curves or loops with it. However, all players share the same tantrix and alternate adding tiles to it. The rules for tile placement are more complicated than for the puzzles, involving a 3-step process in which one player can conceivably add dozens of tiles to the tantrix on a single turn. Further information on the multi-player game is available at the Tantrix web site.

This paper focuses on the puzzles, of which there are many. However, most fall into one of the

4 categories below. Tantrix puzzles have been proven NP-complete via a reduction of the circuit-synthesis problem to a Tantrix task [8].

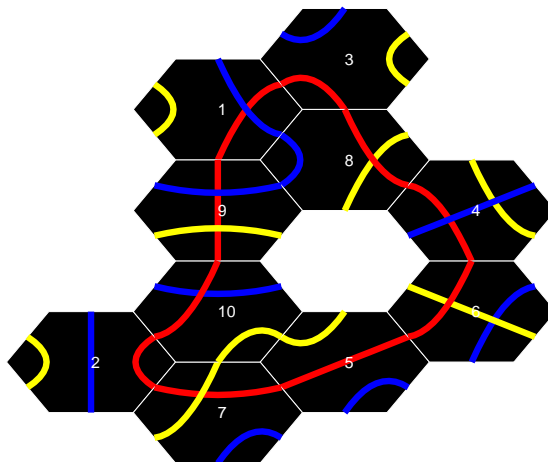


Figure 2: An invalid solution to the 10-puzzle, due to the hole.

3.1 Discovery Puzzles

These puzzles are the simplest to describe but vary in complexity from trivial to extreme. They involve the tiles numbered 1-30. For the k -tile ($3 \leq k \leq 30$) Discovery puzzle, the problem is simply to use the tiles numbered 1- k to build a tantrix with a single k -segment loop. The required color of the loop corresponds to the color code of the k th tile. For example, the 10th tile is in the red group, while the 14th is blue.

3.2 Rainbow Puzzles

These involve the 5 color groups and are named after the color of the group. The specifications vary. The green (10-tile) and yellow (12-tile) puzzles require a tantrix containing a loop of 10 (12) green (yellow) arcs. The blue (10-tile) and red (15-tile) puzzles require a pyramid-shaped tantrix containing a non-looping segment of 10 (15) blue (red) arcs. Finally, the white puzzle requires a tantrix containing a loop (in an unspecified color) of the 9 white tiles. For example, Figure 3 shows the solutions to the two pyramid puzzles.

3.3 Super Puzzles

For these puzzles, a diverse set of 10 or 12 tiles are given, and the solution involves one **or two** loops or segments using all arcs of the given color(s). The tile sets, with their official names, are as follows:

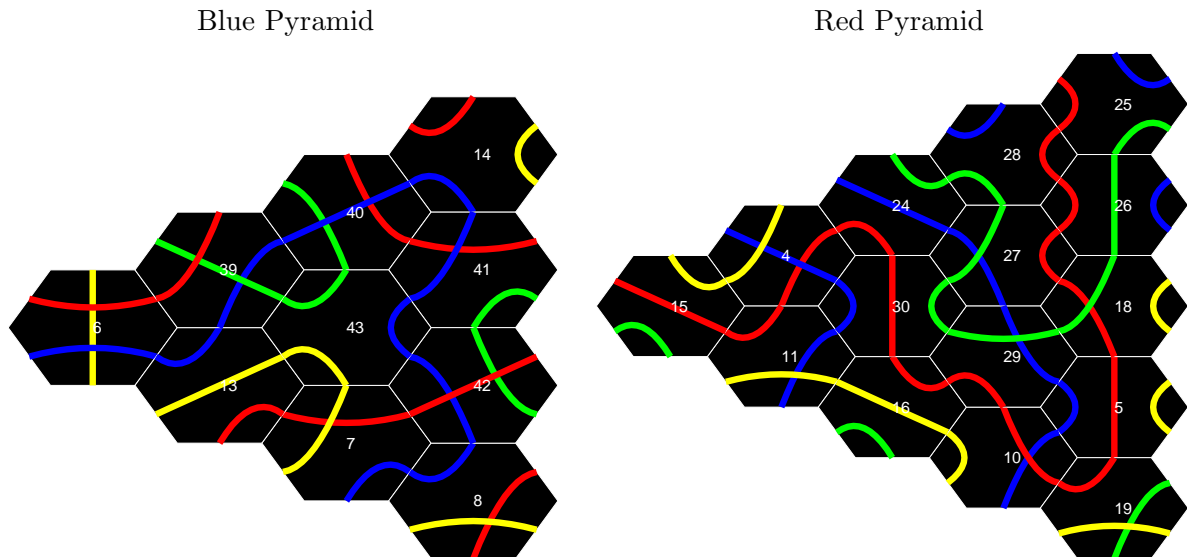


Figure 3: Solutions to the blue and red Rainbow puzzles.

1. Junior (10 tiles) - 3,5,8,12,14,43,46,50,52,54
2. Student (10 tiles) - 19, 21, 24, 25, 29, 31, 32, 40, 41, 42
3. Professor (12 tiles) - 2, 11, 15, 17, 20, 30, 38, 39, 44, 45, 51, 56
4. Master (12 tiles) - 18, 22, 23, 26, 27, 33, 34, 35, 36, 47, 53, 55
5. Genius (12 tiles) - 1, 4, 6, 7, 9, 10, 13, 16, 28, 37, 48, 49

The Junior, Student and Master puzzles require the formation of a single 10- (12-) arc loop in an unspecified color. The Professor puzzle requires the formation of two loops of unspecified colors, where each loop involves all the arcs of its color, while the Genius puzzle requires two non-looping segments using all arcs in two unspecified colors. There are 3 known solutions to the Genius puzzle, two involve red and yellow, and one involves blue and red.

For puzzles in which the colors of the target pattern are not specified in the problem statement, some colors can be eliminated from consideration by a simple arc-counting method. For each color, there are only 3 types of arcs: straight line, sharp 120° turn, and gentle 60° turn. If a set of arcs form a loop, then we can begin at any point on the loop and follow the arcs, updating our current orientation on each new tile. At the end of the loop, the orientation must be the same as at the start. For this to occur, there must be an even number of 60° turns. Hence, any color with an odd number of these gentle turns cannot form a loop with all its arcs. This simple test usually filters out a few colors. For example, it helps in determining that the loop color for the Master puzzle is green. Figure 4 shows one solution.

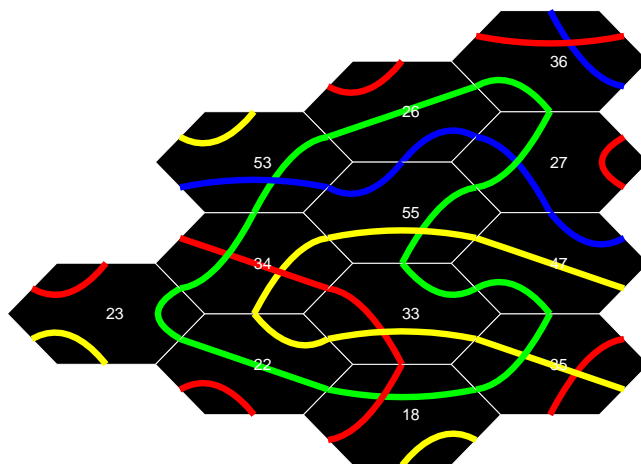


Figure 4: One of a few possible solutions for the Master puzzle.

3.4 Unsolved Puzzles

There are two holy-grail, *Unsolved* Tantrix puzzles, one of which remains unsolved as of this writing. Both involve the entire 56-tile set. The *Curve puzzle* requires a tantrix that contains 4 continuous curves, one in each color, while the *Loop puzzle* requires that the 4 curves be closed.

Each color appears on $\frac{3}{4}$ of the 56 tiles, so there are 42 arcs of each color. Thus, in theory, a perfect solution to the Loop or Curve puzzle would use all 4×42 arcs, yielding a score of 168, where the score is simply the sum of the number of arcs used in the longest segment or loop of each color. However, according to the Tantrix manual, computer analyses have shown that the maximum possible scores for the Curve and Loop puzzles are 146 and 136, respectively.

The manual also reveals that Jack Kuiper solved the Loop puzzle in 2003 (without the aid of a computer), while the best recorded score on the Curve puzzle is 140. Somewhat counterintuitively, in multi-color puzzles, several loops are easier to discover than several open-ended segments.

And if this is not enough of a challenge, Tantrix products include 10 different, but compatible, 10-tile Discovery packs that enable extensions of the basic Discovery puzzles from 30 to 100 tiles.

4 The Evolutionary Computational Challenge/Appeal of Tantrix Puzzles

The combinatorics of Tantrix solutions are quite daunting. First of all, most puzzles involve an unspecified form for the cluster/tantrix; the only constraint is that it is connected and contains no holes. The Golden Rule (i.e., all colors match) greatly restricts the possibilities, but it is difficult to compute its quantitative effect upon this or other contributions to search-space size.

Second, given a pre-defined cluster form, the k tiles must be placed within it and rotated in one of 6 ways. For all except the first tile placed in a cluster, at most 2 of the rotations will match colors with the adjacent tiles in the current tantrix. Hence, given a pre-defined cluster shape, the worst-case size of the search space is:

$$Size_{ss} = 6(k!)2^{k-1} \quad (1)$$

This yields 11147673600 for the 10-tile case and 7653247968377485393920000 for the 20-tile case, and this does not include the combinatorics of the space of legal cluster shapes!

Finally, the fitness landscape (given most straightforward fitness measures) is quite rugged and deceptive. The neighbors formed by swapping any pair of tiles of a $k-1$ tile loop may all be non-loops or even illegal configurations. Although some k -tile solutions stem from small modifications to $k-1$ tile solutions, this is hardly a general rule. In fact, in the Discovery puzzles, the solution to the $k-1$ puzzle may involve a different color entirely from that of the k -tile puzzle.

5 Tantrix-GA

Figure 5 illustrates the conversion from genotype to phenotype (i.e., development) in Tantrix-GA. The chromosome consists of two regions, one for determining the growth pattern of the tantrix cluster, and the other for prioritizing the blocks and selecting their orientations (in cases where several rotations are valid).

The shape region simply encodes a sequence of $k-1$ moves for a k -tile problem, where each move is either South (0), Southeast (1), Northeast (2), North (3), Northwest (4) or Southwest (5). The move dictates the next cell to consider filling; it is always a neighbor of the previously-filled cell.

The second half of the genome consists of k pairs, one for each of the k blocks in the puzzle. The first element of each pair gives a priority to the block - low numbers indicate high priorities - thus determining its placement within the sorted priority list. The second member of the pair is used to choose an orientation for the block in cases where 2 or more orientations would validly match the existing tantrix. For the first block in the list, the tantrix will be empty, so this number will choose one of 6 orientations with which to begin building. All other blocks will have at most 2 feasible orientations. For target patterns with a fixed shape (given by a template to be filled by the tiles), this choice among 6 rotations of the first tile is important. When the shape template is not given, any initial orientation will suffice.

The developmental process is straightforward. The first block is removed from the priority list (which is sorted by ascending priority number) and placed in the middle of the board. The first move is then read from the developmental sequence and the specified neighbor cell becomes the one to fill. The block list is then searched from start to finish until a block that fits into the cell is found. To fit, the block must not only obey The Golden Rule, but it must match up with at least one tantrix edge containing a *focal color*, i.e., a color that is believed to comprise one of the solution

loops. For example, in the blue Rainbow puzzle, the focal color is blue, while in the Genius puzzle, there are 2 focal colors. Development continues until either all blocks are used or the next chosen cell cannot be filled by any remaining blocks.

Let us trace through the development of the 5-tile Discovery puzzle in Figure 5, with red as the focal color. The shape region of the chromosome yields a 4-move sequence: North, Southeast, North, and Northwest. The priorities in the second half of the chromosome dictate the following block ordering: 3,5,1,4,2. Hence, the 3rd block is placed on the board and rotated $5 \cdot 60^\circ$ units clockwise, since 65 is the value of tile 3's orientation gene, and $65 \bmod 6 = 5$. The neutral (0°) orientations are given in the Tantrix manual and are shown for blocks 1-5 in Figure 5, directly beneath their respective chromosomal regions.

Control then moves to the Northern neighbor of tile 3, and tile 5 (next on the priority list) is given the first chance to fill the spot. At this early stage of the solution, two orientations of tile 5 are valid: 0° and 180° clockwise rotations. Either will match the border color(s) to that cell, and in this case, the only border color is the red arc emanating from tile 3's northern edge. Tile 5's orientation gene, 131 is used to choose among these 2 alternatives: $131 \bmod 2 = 1$, so the latter rotation, 180° , wins.

Control then moves to tile 5's Southeast neighbor, which is also tile 3's Northeast neighbor. Tile 1 gets the first chance to fill this spot and successfully does so, but with only one of its rotations, so the disambiguating orientation gene is not needed. The next move is straight North, and tile 4 satisfies this spot with a unique rotation. Finally, tile 4's Northwest neighbor is filled by tile 2 and the puzzle is successfully completed.

The entire tile-placement algorithm is slightly more complicated than shown in the above example. From newly-placed tile T, the shape portion of the genome actually specifies the *first* of T's neighbors, N_1 , to investigate. If N_1 is not on a side that matches a focal-color arc (emanating either from T or from one of the other neighbors of N_1), then the algorithm departs from N_1 and moves counter-clockwise around T in search of the first neighbor, N^* , that does border on a focal-color arc. The building process halts if N^* is not found. Otherwise, N^* , and the remaining neighbors moving counterclockwise from N^* around T, are returned as a set, C. Each remaining tile is then tested for a color match with N^* . If no match is found, the next element in C is tested against all remaining tiles, and so on. This algorithm strongly biases the tantrix growth process toward the formation of long segments and loops in the focal color(s).

Note that nothing in this developmental scheme safeguards against hole creation, nor the formation of long thin segments with no chance of looping back upon themselves. In the multi-person Tantrix game, special move constraints protect against hole formation, while in Tantrix-GA, these restrictions are handled implicitly (and imperfectly) by the fitness function.

5.1 The Fitness Function

In Tantrix GA, fitness assessment of a tantrix involves 4 factors:

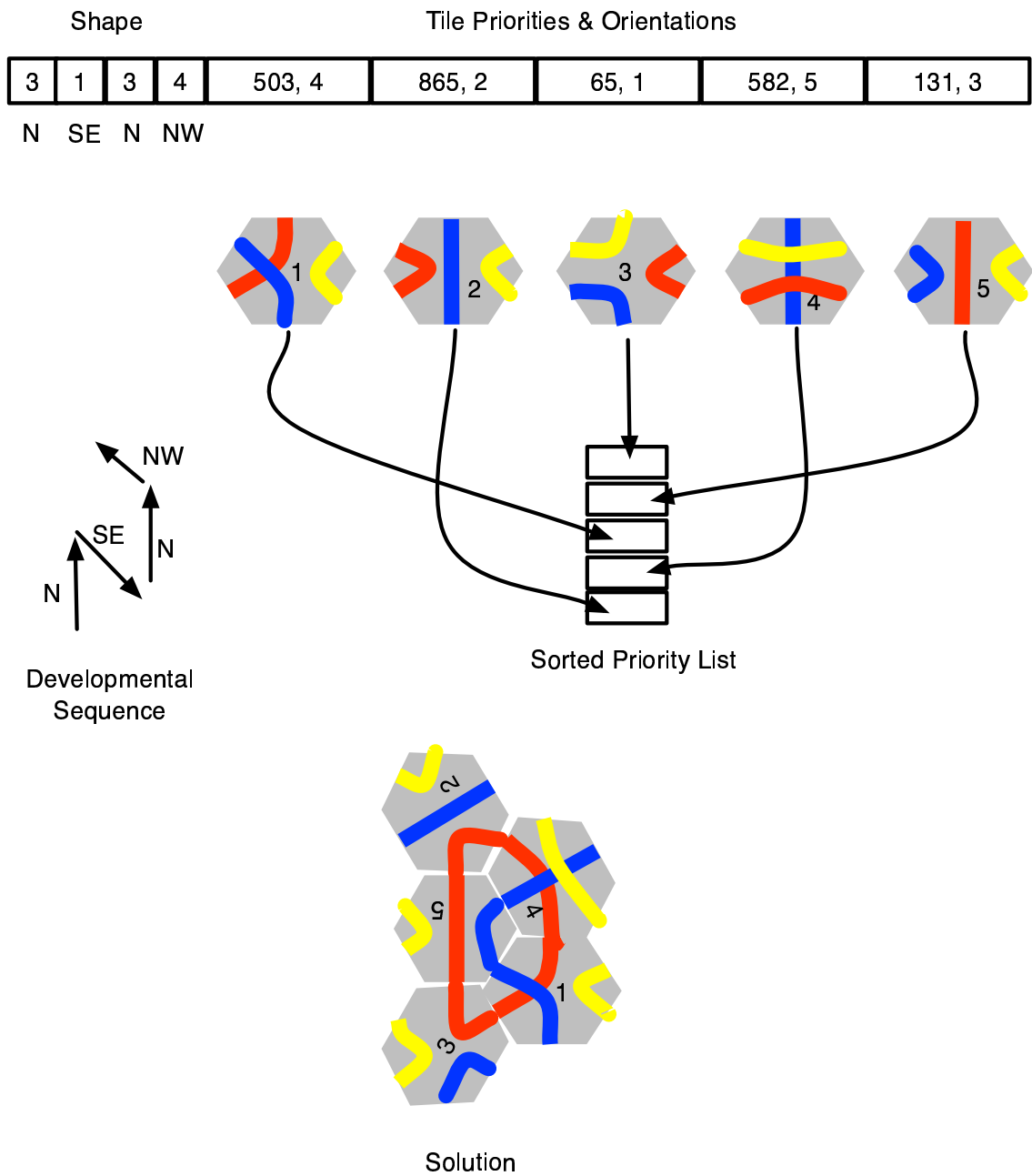


Figure 5: Development of a solution to the 5-tile Discovery puzzle from genotype to phenotype. The tiles labeled 1-5 are the first 5 in the Tantrix Game Pack. The rightmost part of the chromosome determines the priority ordering of these tiles and their preferred orientations, while the leftmost segment defines the overall shape of the tantrix via a growth sequence.

1. Segments (S)
2. Cycles (C)
3. Compactness (M)
4. Holes (H)

For each of the focal colors, the algorithm counts the longest segment and longest cycle in that color. Then either both, or their maximum, become addends for the fitness function. Compactness is simply the average number of tiles surrounding each tile. It is positively weighted in the fitness function to discourage long thin phenotypes, which, typically, cannot involve long loops unless they take wide turns and leave big holes in the middle of the tantrix. Any open cell in the interior of the tantrix constitutes a hole and is penalized severely.

The complete fitness function appears in equation 2:

$$F_{sum} = k_{co}M - k_{ho}H + \sum_{i=1}^{N_{fc}} k_{seg}S_i + k_{cyc}C_i \quad (2)$$

where $1 \dots N_{fc}$ are indices of the focal colors, and S_i and C_i are the largest segment and cycle, respectively, in focal-color i . M is the average compactness of all tiles, and H denotes the number of holes in the tantrix. Unless otherwise stated, the evolutionary runs presented in this article used the following parameter values: $k_{co} = k_{seg} = k_{cyc} = 1$, and $k_{ho} = 10$.

F_{sum} denotes the fact that sizes of the largest segment and cycle are summed for each focal color. An alternate fitness measure, F_{max} (equation 3) used for the 12-piece, two-colored Super puzzles, takes the maximum of the segment and cycle contributions for each focal color.

$$F_{max} = k_{co}M - k_{ho}H + \sum_{i=1}^{N_{fc}} \max(k_{seg}S_i, k_{cyc}C_i) \quad (3)$$

In this case, k_{cyc} should exceed k_{seg} in order to favor cycles over non-cyclic segments. We use $k_{cyc} = 1.5$ and $k_{seg} = 1$ for the Professor puzzle, and $k_{cyc} = 0$ and $k_{seg} = 1$ for the Genius puzzle, which requires non-looping solutions.

In both fitness functions, cycles should not be favored over simple segments to too large a degree, otherwise small (i.e., sub-optimal) cycles dominate the solutions. Ideally, long, convoluted segments evolve and eventually loop back upon themselves.

5.2 Genetic Operators

Tantrix chromosomes are subjected to a variety of genetic operators. First, standard bit-flipping mutation and single-point crossover are employed, with all crossover points restricted to gene bound-

aries. In addition, Tantrix GA uses *headless chicken crossover* [10, 1], wherein parents are occasionally crossed over with randomly-generated individuals. This often functions as a macro-mutation. In the runs reported below, the (bit-wise) mutation rate is 0.01, and the crossover rate is 0.5, wherein 10% of these are the headless-chicken variety.

Two specialized operators are also included: inversion and priority-swapping. During inversion, a random-length sequence of shape genes (the first part of the chromosome) is sliced out of its current location and spliced into a random new location within the shape portion of the same chromosome. This permits basic spatial patterns to change locations within a tantrix. Inversion is performed on child chromosomes with probability p_{inv} , which has a value of .02 in the runs reported herein.

Priority swapping simply exchanges the alleles of two priority genes. However, only certain types of genes can swap alleles. This process mimics a common strategy used by humans to solve Tantrix puzzles: tiles with the same arc angle for a focal color are swapped. At the beginning of a run, 3 priority-swapping bins are formed, one for each of the 3 possible subtended arc angles: 60° , 120° , 180° . Tiles are then placed in all bins for which they have a focal-colored arc of the corresponding angle. Then, during priority-swapping mutation, only genes representing tiles that share a bin can swap alleles. The probability of applying a swapping mutation to a child chromosome, p_{swap} , takes a value of .2 in the simulations reported below. Also, after deciding to swap, Tantrix GA may perform a random number of swaps from the uniform distribution $1 \dots N_{sw}$, where $N_{sw} = 3$ in the reported simulations.

5.3 Selection Procedure

During each generation, all genotypes are converted into phenotypes and evaluated for fitness, with the worst 50% of the population being removed. Those that remain are subjected to sigma-scale selection [16], which scales fitness values by their standard deviation. This helps to combat premature convergence to suboptimal solutions.

A small fraction of each generation stems from elitism, wherein the top 5% genotypically-distinct individuals are copied, without mutation, to the next generation. 85% of the next generation is formed by crossover and mutation of the top 50% from the previous generation, while the remaining 10% comes from randomly-generated individuals. This continuous re-injection helps to avoid convergence in a search space where a) many randomly-generated individuals have reasonable fitness, and b) small modifications to good solutions are often lethal. Basically, many useful genotypes drop out due to mutation and crossover, so random replenishment helps to maintain a viable gene pool.

6 Results

Tantrix-GA runs were performed on all Discovery, Rainbow and Super puzzles. Solutions were found for each; every tantrix figure in this document was discovered by Tantrix-GA. Although the algorithm performs well on most of the smaller puzzles and manages to find multiple solutions for

the 30-tile Discovery puzzle, it has major problems with the 12-piece *Genius* puzzle, which, to date, it has only solved once (in hundreds of attempts).

6.1 Solving Discovery Puzzles

For the Discovery puzzles of 20 tiles or less, Tantrix-GA with a population size of 100 and a maximum of 100 generations consistently discovers solutions. Beyond 20 tiles, larger populations and generations are required, as shown in Table 1. Still, it manages to find 4 (completely different) solutions to the 30-tile puzzle in 20 attempts. Unlike the *Genius* puzzle, which has only 3 known solutions, the many-tiled Discovery puzzles have several solutions. However, they are difficult to find due to many misleading local optima in the search space.

Figure 8 shows solutions to the 26- to 29-tile puzzles. Note that none is a simple modification of the other. Figures 9 and 10 show the sequence of landmark phenotypes (i.e. those having higher fitness than any predecessor) in a successful run on the 30-puzzle. Although the phenotypes show few relationships to one another in early phases, from generation 55 and beyond, the figure traced out by the red curve resembles a growing embryo with a blue *eye* (formed in generation 13) and *mouth* (first appearing in generation 55). The different stages share many groups of adjacent blocks but also vary significantly. The solution is nearly formed at generation 88, where 29 pieces are used, but not until generation 230 are the proper transformations discovered. The step from generation 230 to 241 merely improves fitness by increasing compactness.

To illustrate the difficulty of the search space, at many times during the simulation, the best-of-generation individual is tested by analyzing all immediate neighbors in genotype space (i.e., those within a 1-bit Hamming distance). In almost all cases, the neighborhood in the fitness landscape resembles a plateau with many small cracks (leading to precipitous drops in fitness), as shown in Figure 6. Most such cracks map to the high-order bits of the priority genes. In the Discovery problems, fewer cracks map to the shape genes, although these become more essential for the multi-segment/loop targets in the Super puzzles. Most importantly, there are no spikes leading upwards. Only when genotypes are tested in very early generations does the occasional upward spike appear. Hence, the fitness landscape appears full of high plateaus that are very hard to hill-climb toward.

As a more standard measure of landscape ruggedness, Figure 7 provides a scatter plot of pairs $(\Delta G, \Delta F)$: differences in genotype (Hamming distance) versus differences in fitness. Here, the best-of-generation individual, B, is compared to 5000 randomly-generated genotypes that are between 1 and 100 bit mutations away from B, with 50 genotypes generated for each mutation/Hamming distance. This example yields a Pearson correlation coefficient [4] just over 0.2, and the plot clearly shows no signs of a correlation. Several similar tests during different Tantrix-GA runs on different puzzles yield the same general result: a correlation coefficient between 0.1 and 0.35 but no visible relationship between ΔG and ΔF . In short, the fitness landscapes appear quite rugged.

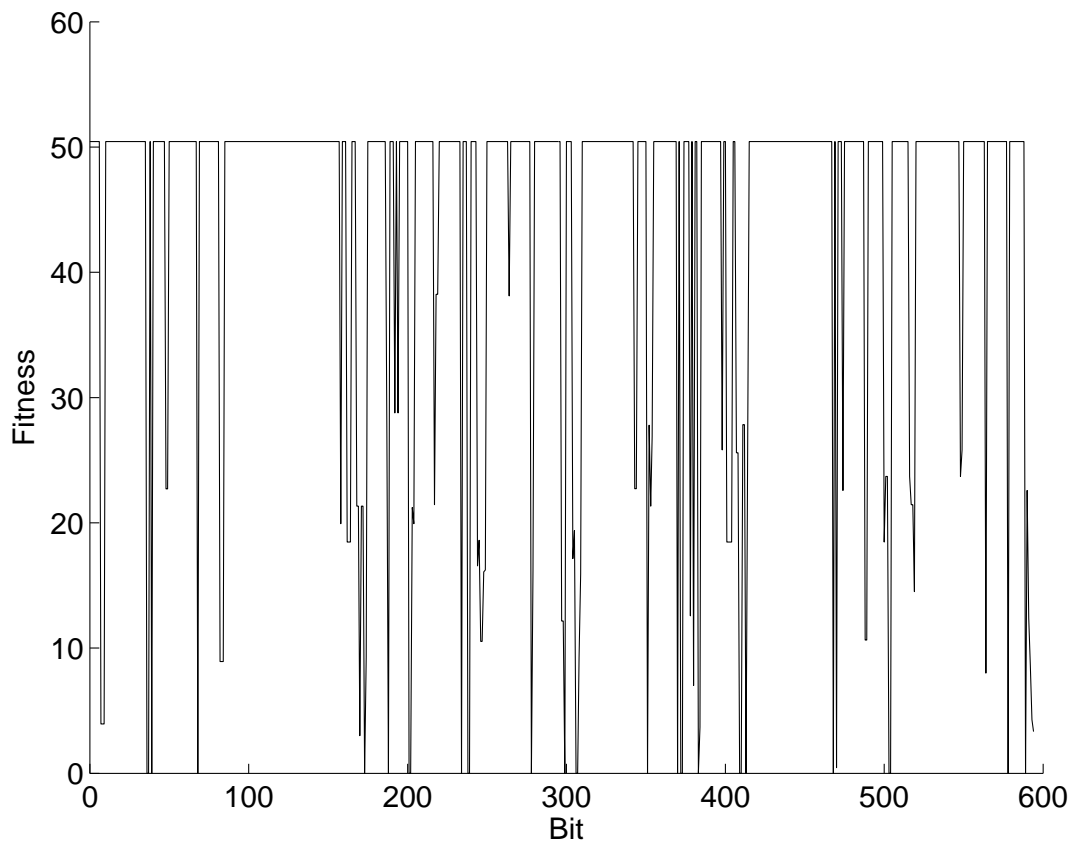


Figure 6: Fitness of all genotypes within a Hamming distance of 1 from a best-of-generation individual during a run of Tantrix-GA on the 30-tile Discovery puzzle. Note the increased sensitivity to mutations of the priority and orientation bits, which begin at location 150, while the shape bits (0-149) are less significant.

Puzzle	Population Size	Generation Limit	Trials	Number Solved	Average Solution Generation
10-tile	100	100	20	20	5.5
12-tile	100	100	20	20	11.7
15-tile	100	100	20	20	20.8
20-tile	100	100	20	4	51.0
20-tile	200	200	20	15	78.9
25-tile	200	200	20	4	142.2
25-tile	300	300	20	6	148.0
30-tile	500	300	20	4	186.8

Table 1: Summary of multiple Tantrix-GA runs on assorted *Discovery* puzzles.

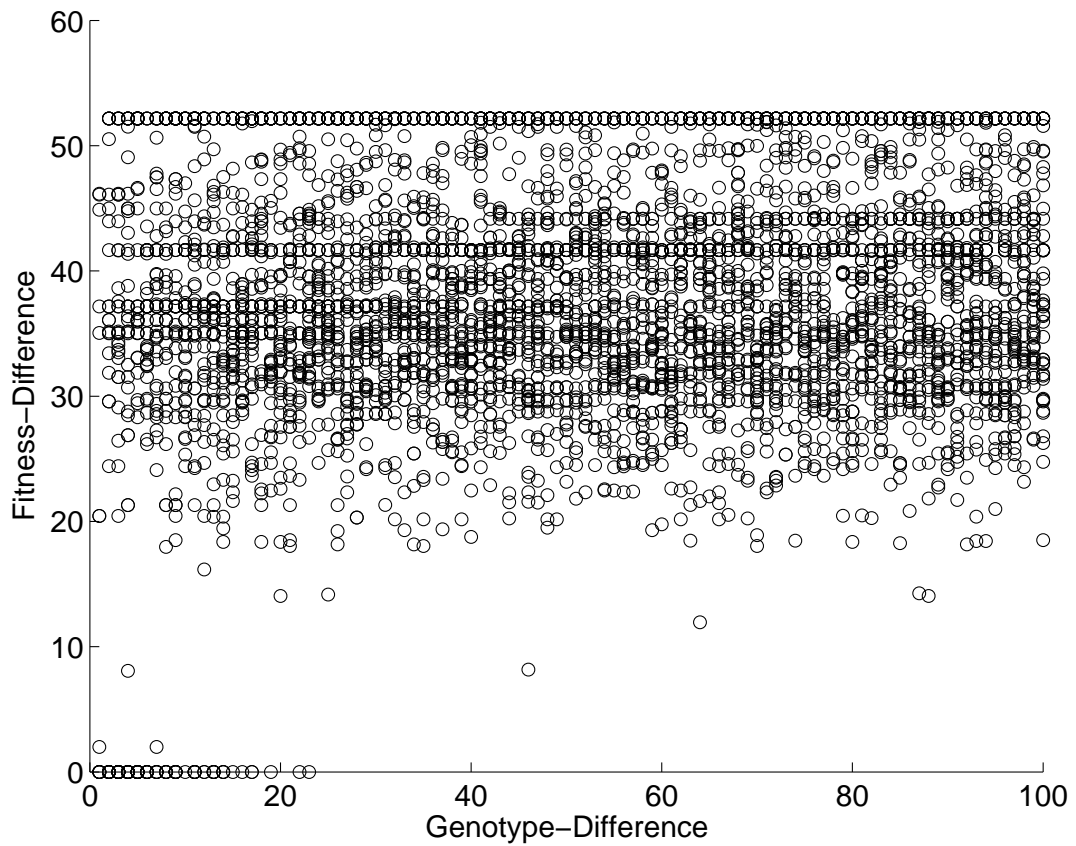


Figure 7: Relationship between genotypic Hamming distance and fitness difference. Given B , the best-of-generation individual (fitness = 52.2) after 100 generations of a run of Tantrix-GA on the 30-tile Discovery puzzle, this looks at mutation classes about B involving 1 to 100 bits, with 50 random samples taken from each class. For each sample, S , the fitness difference between S and B is plotted as a function of the Hamming distance between S and B . All genotypes for this run have a total length of 595 bits. The Pearson correlation coefficient is 0.22

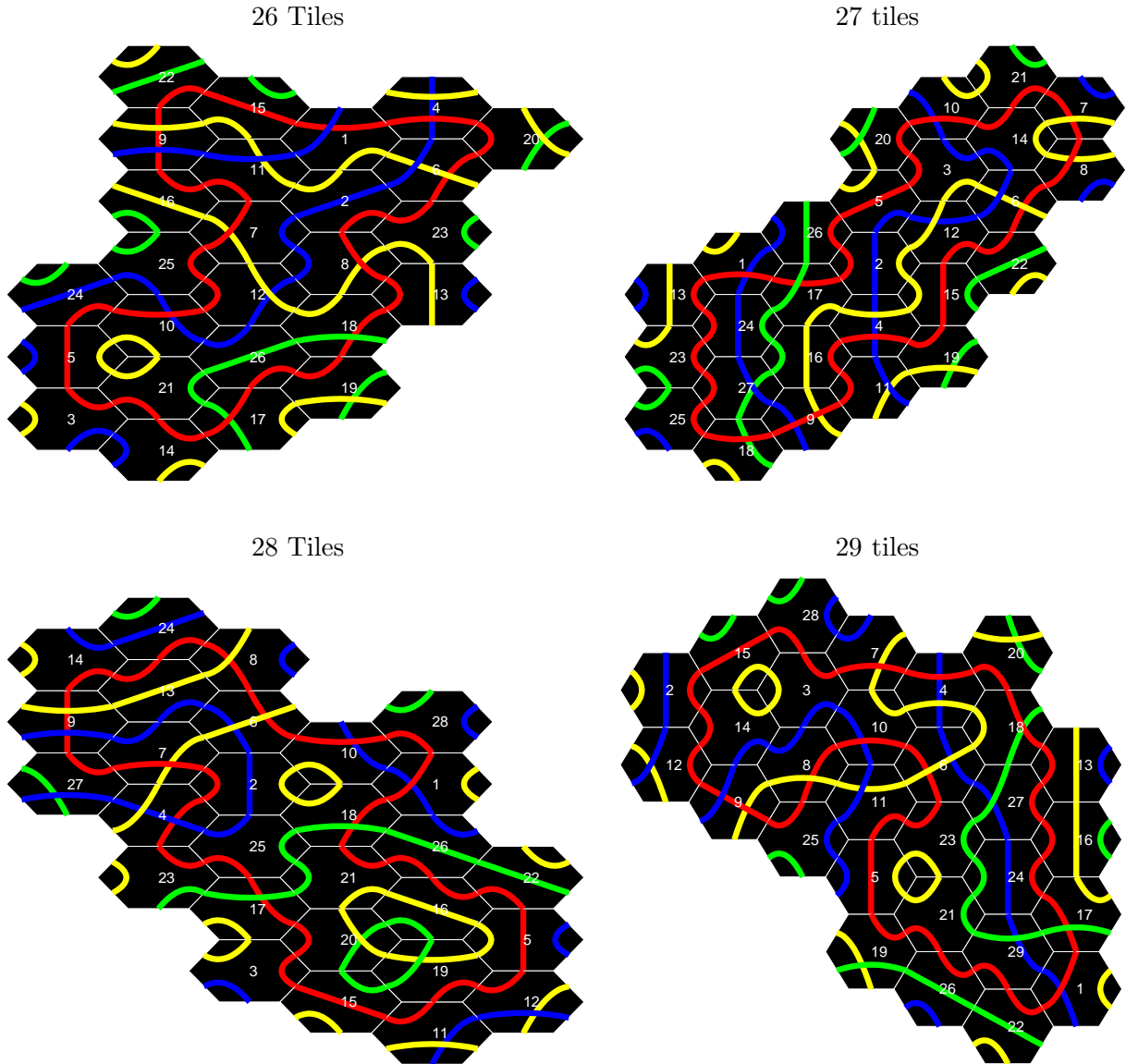


Figure 8: Solutions found by Tantrix GA for the 26- to 29-tile puzzles, all having red as the focal color.

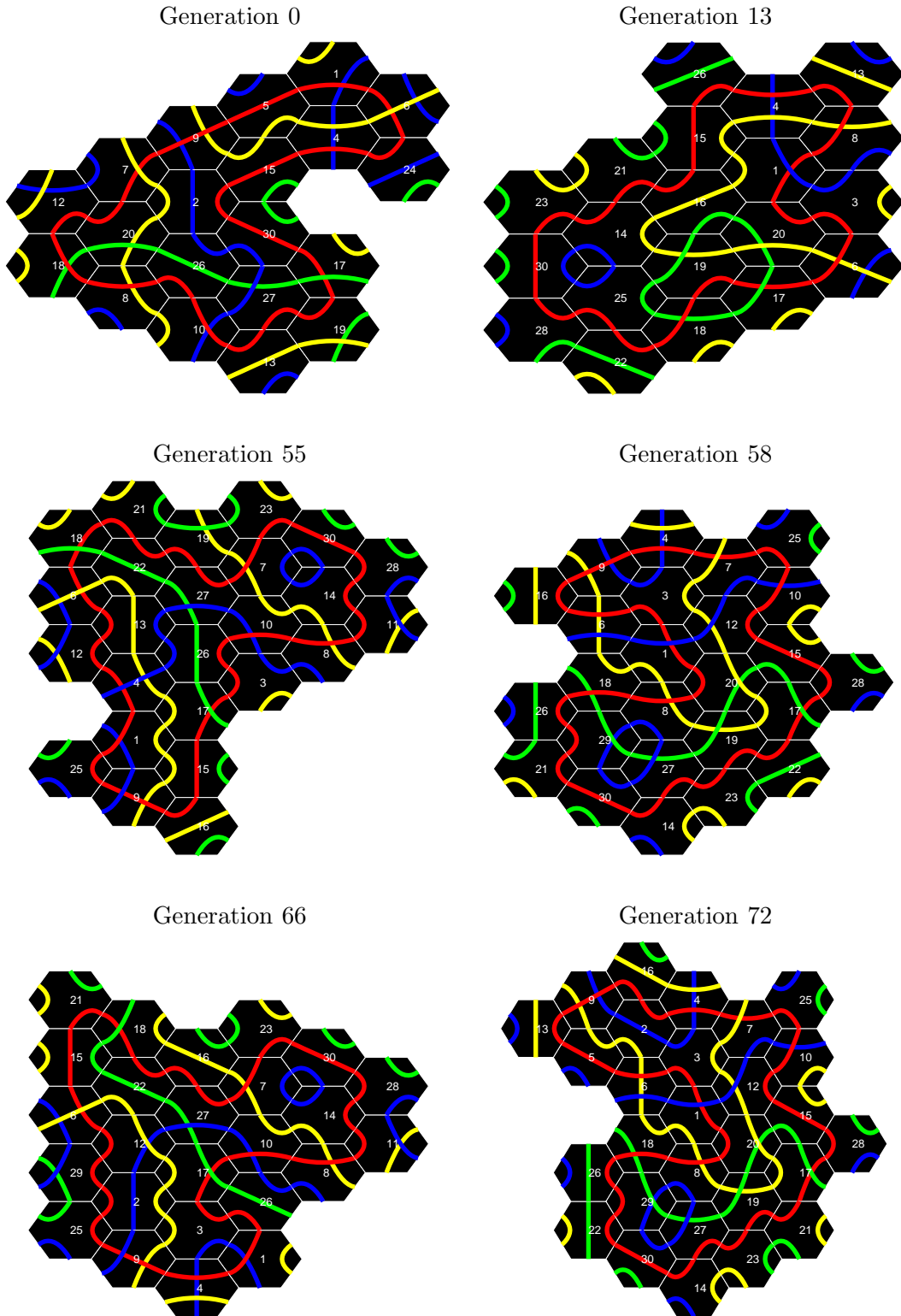


Figure 9: Early stages of the evolutionary progression for the 30-piece Discovery puzzle. The best-of-generation phenotypes for most of the landmark generations are shown. The population size is 500.

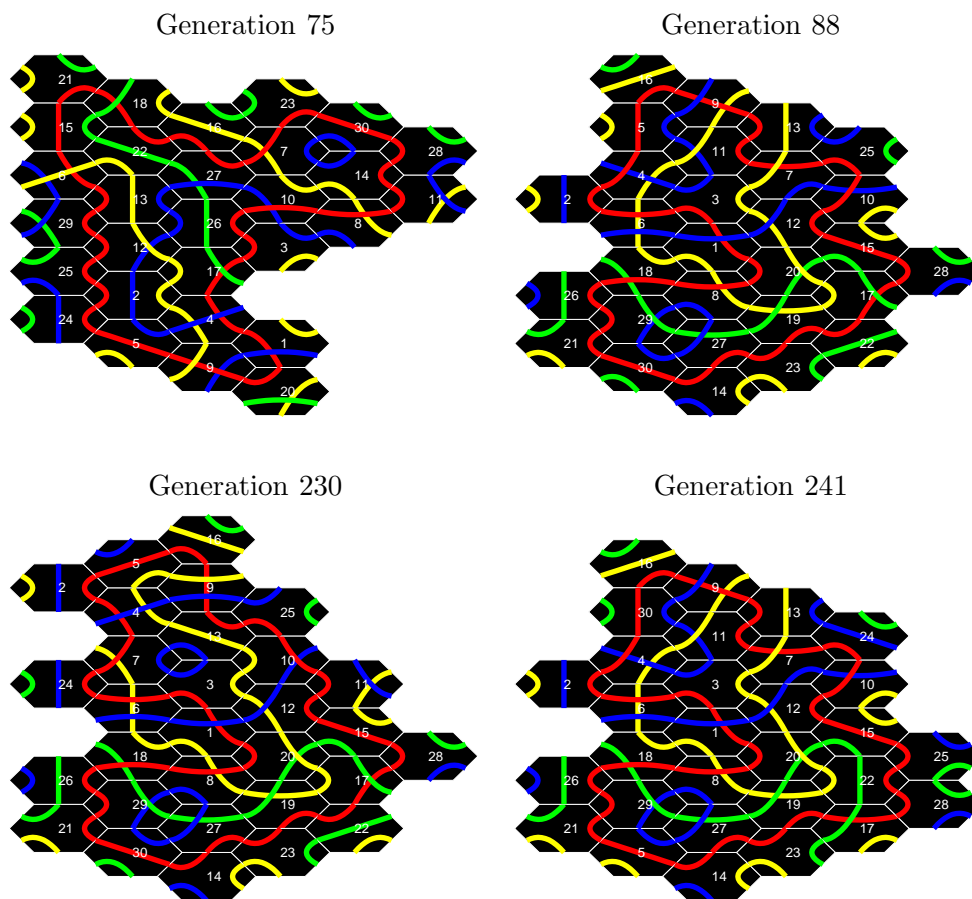


Figure 10: Later stages of the evolutionary progression for the 30-piece Discovery puzzle, again showing only the landmark phenotypes.

Puzzle	Target	Population Size	Generation Limit	Trials	Number Solved	Average Solution Generation
Green	10-loop	100	100	20	17	25.9
Yellow	12-loop	100	100	20	17	54.2
White	9-loop (blue)	100	100	20	20	1.6
Blue	10-pyramid	200	200	20	17	67.7
Red	15-pyramid	200	200	20	2	72.0

Table 2: Summary of multiple Tantrix-GA runs on the 5 Rainbow puzzles. The focal color is the same as the name of the puzzle, except for the White puzzle. Both pyramid puzzles have non-looping target segments.

6.2 Solving Rainbow Puzzles

These 5 puzzles vary considerably in degree of difficulty. The White puzzle is trivial, while the Red puzzle (and the author’s frustration in repeatedly *almost* solving it by hand) was the original impetus for Tantrix-GA. As shown in Table 2, the system has no problem with the first 4 puzzles but also meets its match with the Red puzzle.

The pyramid puzzles are the only ones with a pre-defined topology. Hence, no shape genes are required and the cell-filling algorithm can simply move through the cells of the pyramid in a fixed order, always taking the next tile in the priority list that legally fits the current cell. These pyramids are the only puzzles where a more direct-encoding genome has potential utility; but, as shown later, performance decreases significantly with the more direct scheme.

6.3 Solving Super Puzzles

The Super puzzles pose the greatest challenges relative to their size. None exceeds 12 tiles, but the paucity of solutions and the rugged, deceptive landscapes make them much more difficult than Discovery puzzles of comparable (and larger) size. For example, in comparing Tables 1 and 3, note that the 12-tile Discovery puzzle requires an average of 11.7 generations (rightmost column) to find a solution, while the 12-tile Master puzzle needs 34.1 generations. Similarly, the 10-tile Discovery puzzle uses only 5.5 generations, on average, while the 10-tiled Junior and Student puzzles need 15.2 and 10.6 generations, respectively. Still, for unknown reasons, none of the single-color Super puzzles were as difficult for Tantrix GA as the Yellow Rainbow puzzle (Table 2), which required 54.2 generations, on average.

The two-colored Professor and Genius puzzles immediately compound the complexity and further scramble the search space. For example, in comparing Generations 64 and 106 of the Professor solution in Figure 12, note that the yellow loop involves nearly the same tiles in both cases - as it must since there are only 9 yellow tiles in the puzzle - but tracing clockwise around each loop from tile 45 yields the following tile sequences:

Puzzle	Population Size	Generation Limit	Trials	Number Solved	Average Solution Generation
Junior	100	100	20	16	15.2
Student	100	100	20	18	10.6
Master	100	100	20	18	34.1
Professor	200	200	20	9	74.1
Genius	300	300	20	0	-

Table 3: Summary of multiple Tantrix-GA runs on the 5 *Super* puzzles.

Generation 64: 45, 20, 51, 17, 11, 44, 2, 56

Generation 106: 45, 15, 11, 56, 2, 51, 44, 20, 17

Notice that the only adjacency pair common to both sequences is (2, 56); all others are scrambled. Apparently, evolutionary progress required a dramatic reconfiguration. The evolved solution to the Genius puzzle in Figure 13 shows similar discontinuities along both the yellow and red lines between generations 145 and 192. The fact that Tantrix-GA has solved all other puzzles repeatedly but has only once stumbled upon a Genius solution indicates the needle-in-a-haystack nature of that search space. The fitness progressions for the Professor and Genius puzzles are shown in Figure 11. These are quite typical of Tantrix-GA runs in terms of their step-wise, punctuated equilibria.

7 Alternate Representations and Strategies

The quest for improved search efficiency on the Red Rainbow puzzle, the large Discovery puzzles, and the Genius puzzle inspired a variety of alternate genome representations, genotype-phenotype mappings, fitness functions and selection strategies. Unfortunately, none yielded noticeable improvement, and several were largely disastrous.

7.1 Relaxing The Golden Rule

The Tantrix manual recommends solving puzzles by initially ignoring The Golden Rule and focusing on composing the desired focal-colored loop. Once formed, tiles with equivalent arcs in the focal color can be swapped until The Golden Rule is eventually satisfied.

To incorporate this possibility into Tantrix-GA, the concept of *slack* was defined as the number of mismatches that a tile could have with its neighboring arcs and still be considered legally deployed. A mismatch penalty was added to both fitness functions (F_{sum} and F_{max}) so that tantrices with mismatches scored worse than those without. With $slack > 0$, Tantrix-GA could easily compose long-looped clusters with many mismatches, but coming up with the proper tile swaps (if they even existed) to remove the mismatches proved nearly impossible. Adding slack seems to have only increased the size of the search space without providing any scaffolding that could be exploited by

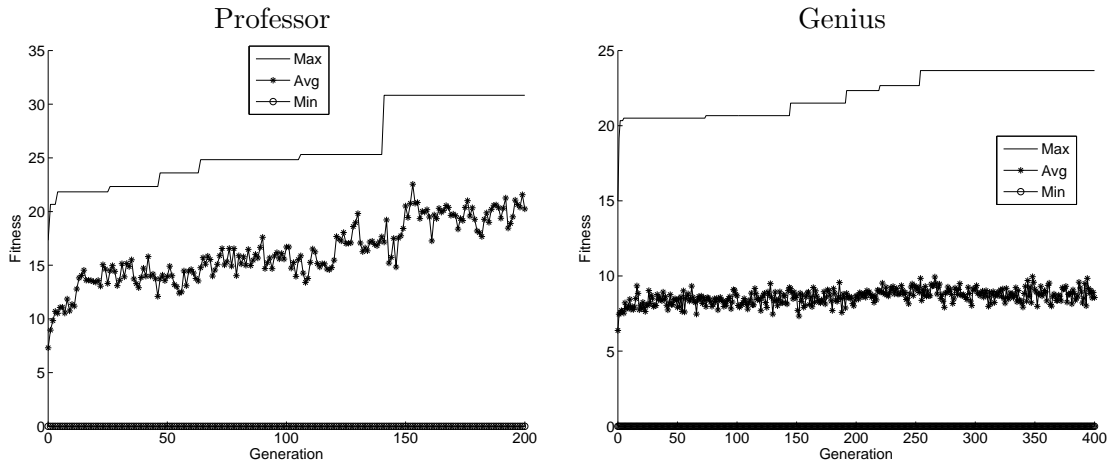


Figure 11: Fitness plots for runs of the *Professor* and *Genius* super puzzles, with populations of 100 and 200, respectively, using the F_{max} fitness function of equation 3.

our genome or genotype-phenotype mapping. Future work could involve changes to these latter elements to better accommodate search in a space dominated by illegal (but potentially helpful) clusters.

7.2 A More Direct Representation

Slack also plays a role in a more direct form of genotype-phenotype mapping that, at least theoretically, could be employed to solve fixed-topology problems such as the pyramid puzzles. In these cases, the shape genes are unnecessary: a fixed sequence of cells in the pyramid can simply be filled in order by the priority-sorted tile genes. As explained earlier, Tantrix-GA’s standard approach simply uses the first tile in the priority list that legally fills a cell (and matches a focal color arc), then it moves on to the next cell in the list. A more direct-encoded solution simply pairs up the cell list and the prioritized tile list such that the k th member of the tile list is always placed in the k th cell, and then oriented according to its rotation gene. Of course, this leads to mismatches, which are penalized by the fitness function. Unfortunately, by expanding the phenotype space to include this multitude of illegal solutions, the direct encoding only seems to exacerbate the search problem.

In effect, the direct encoding forces the use of $\text{slack} = 6$, although in practice the effects of all slack values of 3 or more are the same, since the filling algorithm for pyramids never places a new tile in a cell with more than 3 occupied neighbors; hence a maximum of 3 mismatches are possible at tile-placement time.

Figure 14 illustrates the effects of slack on the fitness landscape. Each graph depicts the fitness of all genotypes a Hamming distance of 1 from the best-of-generation individual. Note that with Tantrix-GA’s standard (indirect) coding and no slack, there are many flat plateaus, indicating neutral local landscapes. Provided that these plateaus are not too large, they can be advantageous for evolutionary search [12, 19, 14, 15]. With the addition of more slack, the plateaus shrink. Since

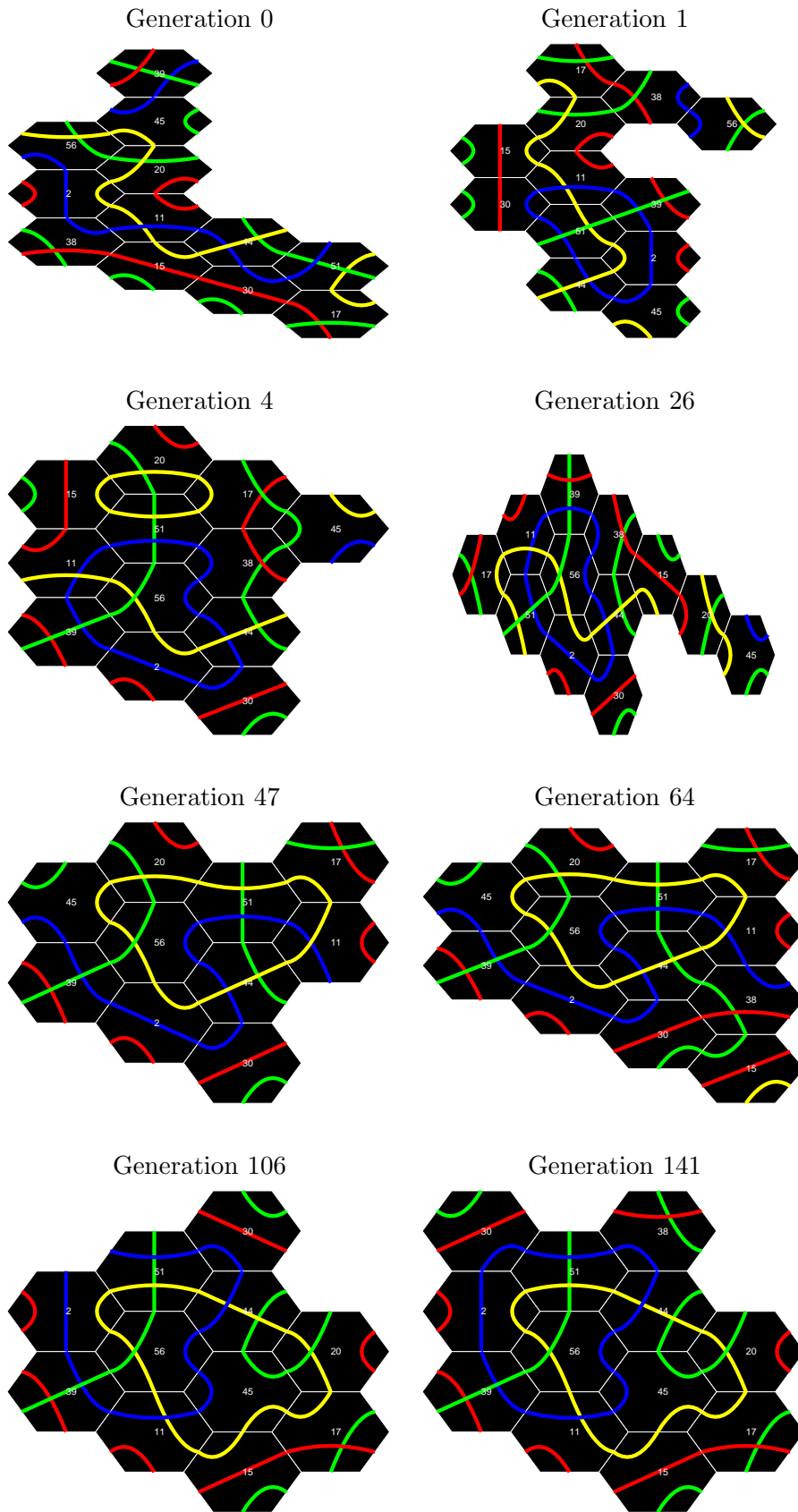


Figure 12: Evolutionary Progression for the 12-piece *Professor* puzzle, which requires both a yellow and a blue loop. The best-of-generation phenotypes for each landmark generation are shown.

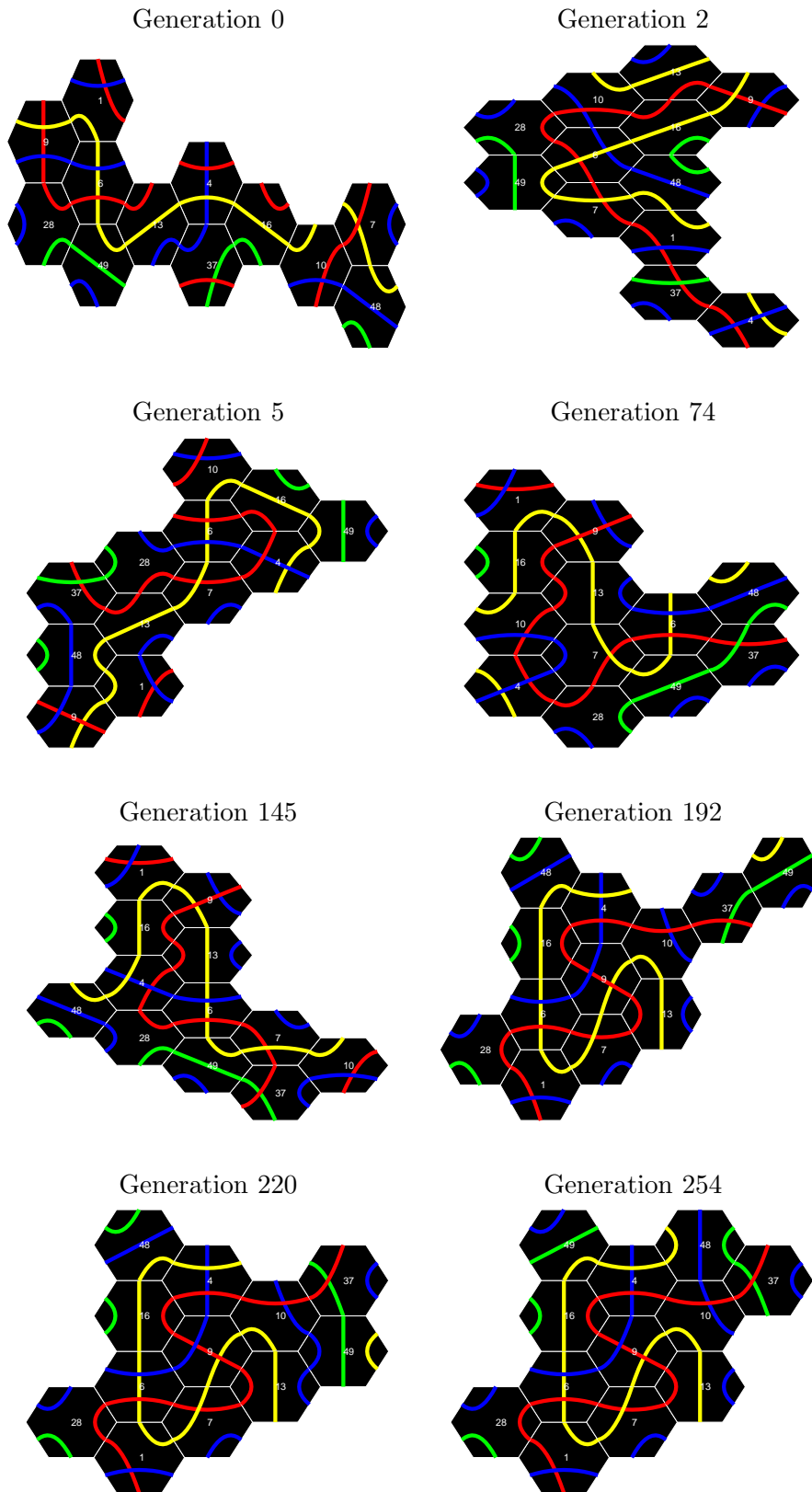


Figure 13: Evolutionary Progression for the 12-piece *Genius* puzzle, which requires both a yellow and a red non-looping segment. The best-of-generation phenotypes for most landmark generations are shown.

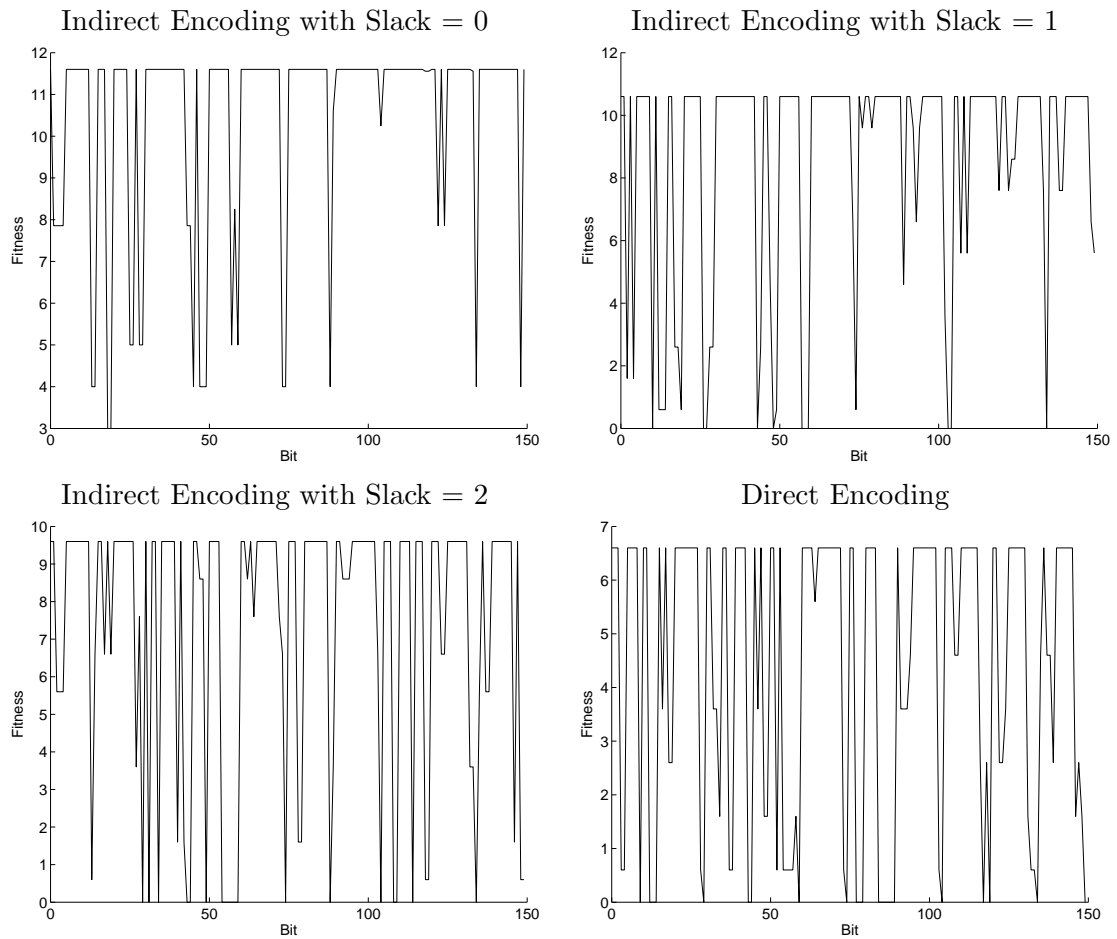


Figure 14: Comparison of fitness landscapes around the best-of-generation individual after 100 generations with a population size of 100 on the 10-tile Blue pyramid puzzle.

direct coding necessitates high slack, it produces difficult, low-neutrality landscapes.

Furthermore, the landscapes appear to become more rugged with the addition of slack. To see this, we can compare runs with and without slack on the 30-tile Discovery puzzle. In the $(\Delta G, \Delta F)$ scatter plot of Figure 15, from the run using slack = 4, notice that the majority of points are proportionately much higher on the y axis than on a similar run without slack, shown in Figure 7. Hence, the sensitivity of fitness to a change in genotype is even stronger when using slack. However, note that there is a slight correlation (upward sloping trend) within the first 10-15 mutations (along the x axis), but even in this region, a small change in the genotype has large fitness consequences.

In general, runs using slack > 0 rarely find solutions to even the simple puzzles. Slack appears to increase ruggedness and decrease neutrality. It may also increase the deceptiveness of landscapes, since it adds more legal partial solutions that may *appear* to be just a few tile swaps away from perfection, but are not. Although this may not rule out the use of all direct representations for Tantrix puzzles, the developmental approaches seem to show greater promise, particularly when cluster shapes are not pre-determined.

7.3 Learning and Lamarckianism

The needle-in-the-haystack nature of certain puzzles motivated the use of local search and partial Lamarckianism [6, 13, 9]. In this scheme, each genotype *learned* by exploring the immediate neighborhood of N 1-swap genotypes (i.e., those formed by swapping 2 priority genes in the original). The fitness of the original was then the fitness of the best such neighbor, and a certain percentage (usually 30) of the learned results were back-coded into the genome prior to reproduction. This only dramatically slowed evolution by adding a factor of N fitness evaluations, and it showed no signs of improved solution-finding on the tough puzzles.

7.4 Genome Simplification

Another potential improvement involves a more efficient chromosomal encoding for the Discovery puzzles. The astute reader may notice that for puzzles involving a single focal color, the Tantrix-GA developmental algorithm is only weakly sensitive to changes in the shape genes. In the developmental algorithm, remember that tiles that can match a focal-color arc are preferred. This means that for single-color puzzles, each new tile will normally extend the current focal-color segment. Hence, when the $k+1$ st tile is played, the k th tile will have only one open (i.e. unconnected) focal-color arc end (since the other end will be covered by the $k-1$ st tile). So the shape gene seems superfluous, since there will only be one open neighbor of the k th tile that can provide a focal-color match for the $k+1$ st tile. The only possible exception is when the growing segment loops back to the first tile, which will still have one open end. Then both the k th and first tile will have an open focal-colored thread. However, if the puzzle is solved correctly, then the $k+1$ st tile will match both the k th and first arc to complete the loop.

In general, only the very first tile will have use for a shape gene, since this tile will have two open ends of its focal-color arc immediately after being played. However, since a loop could be built by

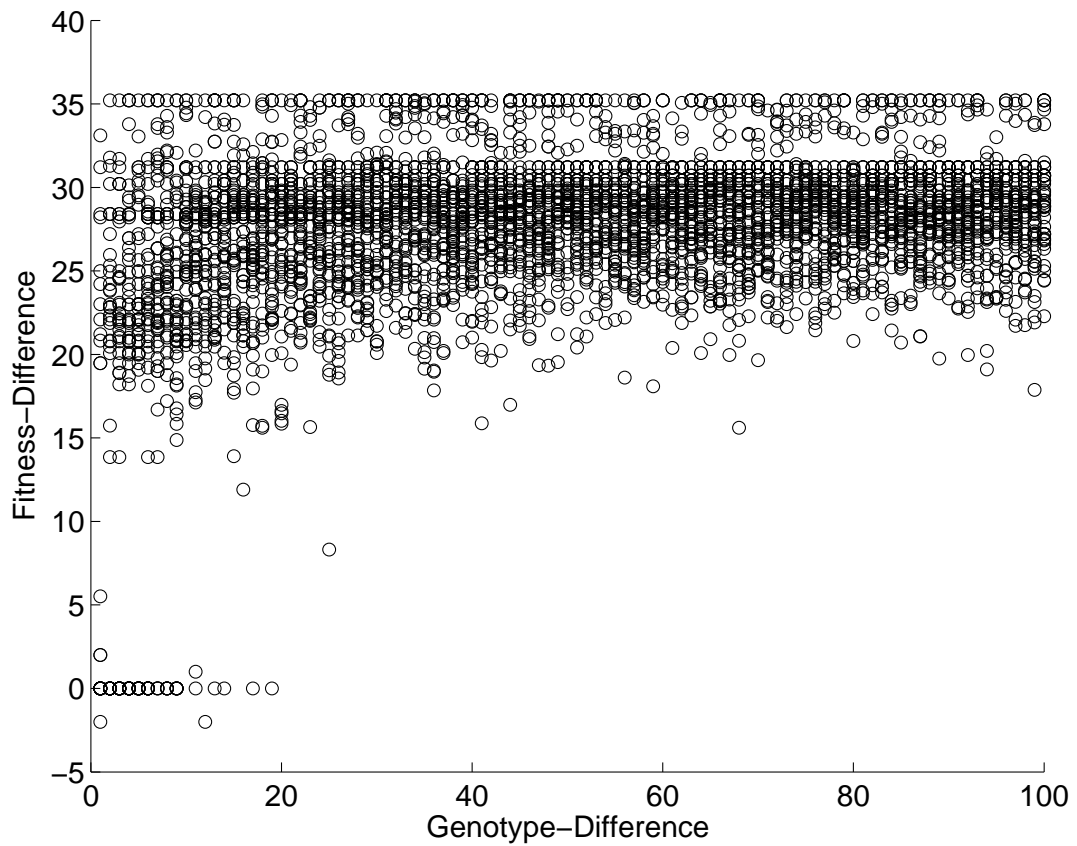


Figure 15: Relationship of genotypic Hamming distance to fitness difference. Given B , the best-of-generation individual (fitness = 35.2) after 100 generations of a run of Tantrix-GA on the 30-tile Discovery puzzle with $\text{slack} = 4$, this looks at mutation classes about B involving 1 to 100 bits, with 50 random samples taken from each class. For each sample, S , the fitness difference between S and B is plotted as a function of the Hamming distance between S and B . All genotypes for this run have a total length of 595 bits. The Pearson correlation coefficient is 0.35.

choosing either one of these ends first, it should be possible to arbitrarily (but consistently) choose one of them and to grow a successful loop in that direction. Thus, even the first shape gene seems superfluous.

As for the orientation genes, they need not encode 6 values, only 2. In Discovery puzzles, the orientation of the initial tile is immaterial, since loops can be grown in any direction, and the hexagonal cell array is, in theory, infinite in both dimensions. Only when the puzzle's shape is pre-determined (as in the pyramid puzzles) is the exact orientation of the first tile critical. So the only potential source of ambiguity is which focal-colored end of the $k+1$ st tile should match up with tile k 's open focal-colored end.

Together, these two observations allow us to greatly simplify the chromosome for Discovery puzzles by removing the shape genes and using a single bit to encode orientations. Only the priority genes remain the same. Although this simplification does not degrade performance, it yields no significant improvement. Quite possibly, the smaller genotype search space is offset by decreased neutrality.

7.5 Diversity Enhancement

Although the reinjection of 10% random individuals in each generation plus the macromutations of headless-chicken crossover helped avoid total convergence, they could not insure that multiple peaks of the fitness landscape were being explored simultaneously. To better provide for a diversity of *good* individuals, as opposed to just diversity, sharing [5] and clearing [17, 18] were introduced into the selection algorithm.

To wit, individuals were sorted into groups based on similarity (of either their raw fitness values or their genotypical bit strings). A fixed number, k (usually 3) individuals in each group were allotted their full normal fitness, F , while the remaining $M-k$ individuals *shared* F ; i.e., each received an adjusted fitness of $F/(M - k)$, where M is the group size. In general, this gave no significant performance improvement. In particular, it did not help in the quest for a consistently effective search for the Genius puzzle solutions.

7.6 Self-Organizing Approaches

Since most of the Discovery puzzles have multiple (if not many) solutions, the entertaining idea of self-organizing solutions seems much more plausible in the Tantrix domain than in games where only unique solutions exist. To investigate this, we devised a Swarm-based Tantrix in which tiles randomly move around the hexagonal grid, with the only restriction being that tiles cannot dock in violation of The Golden Rule.

The model includes a *persistence* state variable for each tile, with high persistence indicating a stochastic tendency to stay put. To encourage the formation of long loops and segments in the focal color, a tile's persistence positively correlates with the number of adjacent tiles having matching focal-colored arcs. Hence, tiles that participate in a focal-color segment are less likely to move.

Although this routinely grows segments of 8-12 focal-colored arcs, it rarely grows loops. Doing so might require more complicated signaling, thus moving well beyond the Swarm philosophy.

Alternatively, the Tantrix blocks could be interpreted as metabolizing organisms in an artificial-life world, with chemical recycling embodied in loop formation. Successful ecosystems would then be those that formed clusters of organisms that recycled several chemicals, i.e., housed many loops in their tantrix. With an additional mechanism for clusters to move as units, collide, etc., the ecosystem could involve clashes among populations, resulting in dissolving, fusing, and in general, reorganizing clusters.

In short, the abstract nature of the Tantrix tiles and the constrained but not overly-restrictive nature of the binding chemistry (i.e., many possible neighborhood combinations are legal) could support a host of interesting examples of emergent, life-like behavior.

7.7 Future Attempts

Tantrix-GA now includes a wide variety of parameters and genetic operators which we have tuned only to the degree that the system can (rather reliably) find solutions to all the above-mentioned problems, except the notorious Genius puzzle. Further work in tuning the parameters to more properly dovetail with the simplified genome and diversity-enhancement selection operators might improve upon the performance in Figure 1, for example, but this in itself is relatively uninteresting. The real profit in such improvements would be to pave the way for solutions to the 40-, 50-, even 90-piece Discovery puzzles.

Different advances will be necessary to attain:

1. Better performance on the Genius puzzle.
2. A more solid basis for attacking the *Unsolved* puzzles.

No obvious representational changes nor genetic operators immediately present themselves. They will probably require a deeper mathematical analysis of these puzzles as the foundation for a complex (possibly non-intuitive) genotypic representation and developmental process that have little connection to human heuristics for puzzle solving.

8 Discussion

The main purpose of this paper is to introduce Tantrix as an interesting problem domain for evolutionary computation. Tantrix-GA is, to the best of our knowledge, the first attempt to solve such puzzles with a genetic algorithm, but future systems will surely achieve higher performance levels. The challenges posed by both the Genius and Unsolved puzzles will hopefully inspire others to delve into this fascinating area.

Although lacking the easily-visualizable mathematical beauty of the De Jong test suite [11], Tantrix clusters have a pleasing graph-theoretic essence that has attracted thousands of players to the game. Also, the wide variety of Tantrix puzzles provides a test suite of its own, which, although lacking the wide conceptual span of De Jong's famous collection, does provide a graded, challenging environment for designing and benchmarking indirect representations.

Similar to much of the work in evolutionary design and creativity [2, 3], evolutionary Tantrix solving appears to profit most from indirect representations. However, it requires very little background knowledge about a complex domain (such as electronics, music, civil engineering, etc.), employs a straightforward fitness metric (that demands no expensive simulations nor ad-hoc quantifications of subjective human criteria), and involves merely a bag of 56 colored tiles, available in most hobby shops.

Thus, Tantrix provides an easily-accessible venue for evolutionary computation teaching and research that has a bit more aesthetic appeal than traveling-salesman or job-shop-scheduling tasks. The Tantrix test suite covers many levels of difficulty, from the simpler Discovery puzzles - solvable by any student with a basic GA and a rudimentary understanding of Tantrix - to the larger Discovery puzzles and multi-colored Super puzzles, which require detailed attention to chromosomal encoding, genetic operators and fitness functions.

Our experience indicates that although several of the most difficult puzzles have needle-in-a-haystack characteristics, and all of the puzzles have a high degree of deceptivity, there are still possibilities for assigning proper partial credit to intermediate solutions. Basically, not all promising partial solutions are deceptive and even deceptive solutions have useful building blocks. Furthermore, many of the puzzles have a multitude of viable solutions, albeit well dispersed and sitting atop sharp plateaus.

The trick(s) seem to be in choosing the fitness function and in biasing the growth process. Of all the modifications made during the design and testing of Tantrix-GA, none had a more dramatic positive effect than the introduction of the matching-focal-color constraint on the placement of the next tile in the growth algorithm. This, combined with the compactness constraint, quickly led to the formation of long, hole-free loops that gradually expanded into complete solutions. In addition, the move to a simpler fitness function that focused on the maximum-length segment and loop (as opposed to the average length of all focal-colored segments and loops) proved very significant.

Beyond these design decisions, however, nothing yielded quantum performance improvements. However, to attack the Unsolved puzzles, a few major conceptual improvements are surely necessary, along with a few orders of magnitude more computing power. We have focused on small populations and short runs in the belief that proper representation is the key to Tantrix solving, not brute computational force. Furthermore, the argument for Tantrix as a useful EC teaching tool loses some credibility if all examples require a 1000-node cluster.

Still, the Unsolved puzzles have complexity levels that dwarf everything solved in this paper, forcing an inevitable divergence of teaching and research interests. To this end, we are currently working on a parallel version of Tantrix GA.

References

- [1] W. BANZHAF, P. NORDIN, R. E. KELLER, AND F. D. FRANCONI, *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*, Morgan Kaufmann, dpunkt.verlag, 1998.
- [2] P. BENTLEY, *Evolutionary Design by Computers*, Morgan Kaufmann Publishers, San Mateo, CA, 1999.
- [3] P. BENTLEY, *Creative Evolutionary Systems*, Morgan Kaufmann Publishers, San Mateo, CA, 2001.
- [4] P. COHEN, *Empirical Methods for Artificial Intelligence*, The MIT Press, Cambridge, MA, 1995.
- [5] D. GOLDBERG, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, Reading, Massachusetts, 1989.
- [6] G. E. HINTON AND S. J. NOWLAN, *How learning can guide evolution*, *Complex Systems*, 1 (1987), pp. 495–502.
- [7] J. H. HOLLAND, *Adaptation in Natural and Artificial Systems*, The MIT Press, Cambridge, MA, 2 ed., 1992.
- [8] M. HOLZER AND W. HOLZER, *Tantrix rotation puzzles are intractable*, *Discrete Applied Mathematics*, 144 (2004), pp. 345–358.
- [9] C. R. HOUCK, J. A. JOINES, M. G. KAY, AND J. R. WILSON, *Empirical investigation of the benefits of partial Lamarckianism*, *Evolutionary Computation*, 5 (1997), pp. 31–60.
- [10] T. JONES, *Crossover, macromutation and population-based search*, in *Proceedings of the Sixth International Conference on Genetic Algorithms*, L. Eshelman, ed., Pittsburgh, PA, Morgan Kaufmann, 1995, pp. 73–80.
- [11] K. D. JONG, *An analysis of the behavior of a class of genetic adaptive systems*, PhD thesis, University of Michigan, Ann Arbor, Michigan, 1975.
- [12] M. KIMURA, *The Neutral Theory of Molecular Evolution*, Cambridge University Press, Boston, MA, 1983.
- [13] J. B. LAMARCK, *Of the influence of the environment on the activities and habits of animals, and the influence of the activities and habits of these living bodies in modifying their organization and structure*, *Zoological Philosophy*, (1914), pp. 106–127.
- [14] M. LONES, *Enzyme Genetic Programming*, PhD thesis, University of York, York, England, 2003.
- [15] M. A. LONES AND A. M. TYRRELL, *Biomimetic representation in enzyme genetic programming*, *Genetic Programming and Evolvable Machines*, 3 (2002), pp. 193–217.
- [16] M. MITCHELL, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, MA, 1996.

- [17] A. PETROWSKI, *A clearing procedure as a niching method for genetic algorithms*, in Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 1996, pp. 798–803.
- [18] B. SARENI AND L. KRAHENBUHL, *Fitness sharing and niching methods revisited*, IEEE Transactions on Evolutionary Computation, 2 (1998), pp. 97–106.
- [19] A. S. WU AND R. LINDSAY, *Empirical studies of the genetic algorithm with non-coding segments*, Evolutionary Computation, 3 (1995), pp. 121–148.