Evolutionary Exploration of the Mandelbrot Set

Daniel Ashlock
Mathematics and Statistics
University of Guelph,
Guelph, Ontario
Canada, N1G 2W1
dashlock@uoguelph.ca

ABSTRACT

The Mandelbrot set is an infinitely complex fractal defined by a simple iterative algorithm operating on the complex numbers. Views of the Mandelbrot set are a common form of fractal art. Presented here is a collection of fitness functions that permit three-parameter evolutionary search of the Mandelbrot set to locate interesting views. While the system presented is automatic, the hand of the artist can direct the type of views found by modifying the fitness function. Based on an envelope that specifies the character of the fractal landscape desired, the fitness function is easily reconfigured with minimal programming skill and without knowledge of complex arithmetic.

I. INTRODUCTION

The Mandelbrot set [8] is a fractal with infinite complexity. A pair of views into the set are shown in Figure 1. One sense in which the Mandelbrot set has infinite complexity is that if a well-chosen portion of the set is rendered, then it possesses new structures not seen at any coarser view. This fact is not central to this paper and is offered without proof but with the following evidence. Examine Figure 1. The “lakes” along the flank of the set have “rivers” that come out of them. The rivers have various branching factors. The largest lake in the finer view in the bottom half of the figure has rivers with a branching factor of 3. The largest lake to its left has rivers with a branching factor of 5. The largest lake between them has rivers with a branching factor of 3+5=8. These lakes are labeled with their corresponding numbers. This pattern continues indefinitely and so branching factors of all degrees in the Fibonacci sequence may be found in this fashion (in fact lakes with all possible positive integer branching factors may be found in infinitely many places).

This paper presents an evolutionary algorithm that searches for potentially interesting views of the Mandelbrot set at a limited but variable level of resolution. The evolutionary algorithm is a simple real-parameter optimizer. The novel feature is a type of fitness function that permits the artist to control, to a degree, the character of views being searched for. Use of an evolutionary algorithm provides a diversity of views as shown in Figures 2, 3, and 4. These figures display 20 evolved views for three different fitness functions of the type presented here. Unlike many optimization efforts, the search

Fig. 1. The entire Mandelbrot set and a zoom on its upper-right flank. Black points are members of the set, other points are colored with shades of gray with darker colors denoting more iterations to escape.
for “interesting” views is served well by the location of local optima in the fitness landscape.

This paper is not the first to evolve fractals, but it is in the rarer of two major categories of such efforts. Because writing a fitness function that can judge if a fractal in interesting is difficult (and not a well-defined problem), the most common sort of fractal evolution is human-in-the-loop evolution in which a human being is used as the fitness function. Examples of human-in-the-loop evolution include systems that use genetic programming [9] and which optimize parameters of (generalized) Mandelbrot sets to generate biomorphs [11]. Fractals that are located by evolutionary real parameter optimization to match pictures of faces appear in [12].

Iterated function system fractals, explained in detail in [7], are the major target of evolution in [10] and were used to perform fractal rendering of DNA sequences in [5]. A hybrid representation using both finite state machines and iterated function systems was evolved to render fractals from different types of DNA in [4] and [6].

L-systems or Lindenmayer systems are grammatical models that can be used as a representation for the evolution of fractals. Grammatical systems start with an initial string. Characters within the string are expanded by the rules of the grammar, iteratively, to obtain a single string. The characters are then interpreted by a renderer, such as a graphic turtle, to yield a fractal. Such evolution of L-systems that are rendered as plants by a graphic turtle is presented in [1], [2]. Fractal L-systems that yield music appear in [3].

The remainder of this paper is structured as follows. Section II gives the mathematical background, introducing complex arithmetic for those who are unfamiliar with it and formally defining the Mandelbrot set and the notion of a view into it. Section III defines the family of fitness functions that form the core of this paper. Section IV gives the experimental design. Section V gives the results in the form of thumbnails of the views discovered by the evolutionary algorithm. Finally, Section VI suggests alternate fitness functions and suggests ways to apply these techniques to other kinds of fractals.

II. COMPLEX ARITHMETIC AND THE MANDELBROT SET

The complex numbers are an extension of the familiar real numbers (those that represent distances or their negatives) achieved by adding in one “missing” number \( i = \sqrt{-1} \) and then closing under addition, subtraction, multiplication, and division by non-zero values. The number \( i \) is called the imaginary number. A complex number \( z \) is of the form \( z = x + yi \) where \( x \) and \( y \) are real values. The number \( x \) is called the real part of \( z \), and \( y \) is called the imaginary part of \( z \). The arithmetic operations for complex numbers work as follows:

\[
\begin{align*}
(a + bi) + (c + di) &= (a + c) + (b + d)i \\
(a + bi) - (c + di) &= (a - c) + (b - d)i \\
(a + bi) \times (c + di) &= (ac - bd) + (ad + bc)i \\
\frac{a + bi}{c + di} &= \frac{ac - bd + bc - ad}{c^2 + d^2} + \frac{bc + ad}{c^2 + d^2}i
\end{align*}
\]

One of the pleasant properties of the complex numbers is that they place an arithmetic structure on points \((x, y)\) in the Cartesian plane so that arithmetic functions over the complex numbers can be thought of as taking points \((x, y)\) (represented by \(x+yi\)) in the plane to other points in the plane. Because the complex numbers have this one-to-one correspondence with the points of the Cartesian plane, they are also sometimes referred to as the complex plane. The Mandelbrot set is easier to define when it is thought of as consisting of points in the plane. The absolute value of a complex number \(z = x + yi\) denoted in the usual fashion \(|z|\) and has as its value the distance \(\sqrt{x^2 + y^2}\) from the origin of the complex plane.

Definition 1: The Mandelbrot set is the collection of points \(z\) in the complex plane for which the sequence:

\[
z_0 = z \\
z_{n+1} = z_n^2 + z
\]

fails to diverge in absolute value. This sequence is referred to as the Mandelbrot sequence for \(z\). In other words, a point \(z\) is in the Mandelbrot set if the absolute value of the members of the Mandelbrot sequence stays bounded. If the sequence stays bounded, then it is in fact bounded by two [8].

For a point not in the Mandelbrot set the divergence behavior of that point is the number of iterations of the Mandelbrot sequence required for a term with absolute value in excess of two to appear. Most of the complex plane has a divergence behavior of 0 as most points start with an absolute value in excess of 2.

Definition 2: A view into the Mandelbrot set is the specification of a region, square in this paper, of the complex plane together with a pixel resolution. The Mandelbrot sequence is sampled, once per pixel, and a rendering of the Mandelbrot set (and the divergence behavior of points outside the set) within the square is computed. These are then colored in some fashion to yield a picture which is called a depiction of the view.

This paper uses a coloring scheme with a uniform shade of gray for points in the Mandelbrot set and a periodic grayscale for divergence behaviors. When checking for membership in the Mandelbrot set it is necessary to set an upper limit \(M_{max}\) on the number of terms of the Mandelbrot sequence checked. In this paper \(M_{max} = 200\). Points that reach 200 terms of the Mandelbrot sequence without achieving an absolute value of two are rendered as if they are within the Mandelbrot set.

III. FITNESS FUNCTION DESIGN

The fitness functions used in this study all sample the Mandelbrot sequence at 121 points arranged in an \(11 \times 11\) grid within the view. This grid is square, equally spaced, and anchored at the corners of the view. The number 11 was chosen by preliminary experimentation: the apparent ability of the fitness function to capture the character of a view improves with the number of sample points while speed declines. The number 11 represents an ad-hoc balance between patience and
quality. The number of iterations, up to $M_{\text{max}}$, at each sample point is computed yielding 121 integers in the range 0 – 200. Fitness is then computed as the sum of squared differences of the sampled values from a mask. This fitness is minimized by the evolutionary algorithm.

Three masks were tested, and their values are given by equations 1, 2, and 3. A mask is simply a set of 121 desired iteration values that the algorithm is trying to match within a view. The constants $u$ and $v$ in the equations used to specify masks in this study are real numbers taking on 11 equally spaced values in the range $-1 \leq u, v \leq 1$. The values are rounded to the nearest integer before use. A mask could simply be specified by filling in desired integer values in an $11 \times 11$ array. Equation 1 is a hill centered in the middle of the mask region; equation 2 is a constant function; equation 3 is a ridge-shaped function running from the upper left to the lower right of the view. The integral values for the non-constant masks are given in Table I.

$$
\text{mask1}(x, y) = M_{\text{max}} \cdot \frac{1.2}{3u^2 + 3v^2 + 1}
$$

$$
\text{mask2}(x, y) = M_{\text{max}} \cdot 0.75
$$

$$
\text{mask3}(x, y) = M_{\text{max}} \cdot \frac{1}{(u - v)^2 + 1.2}
$$

These three masks were chosen because each yields a different character of views. There are a vast number of other possible masks, and researchers or artists interested in experimentation and exploration are invited to contact the author.

IV. SPECIFICATION OF EXPERIMENTS

A view into the Mandelbrot set is specified by a corner $(x, y)$ and side length $l$. The evolutionary algorithm operates on a collection of 800 such three-parameter views. The initial population is generated by placing $(x, y)$ uniformly at random in the square region with corners $(-2, -1.5)$ and $(1, 1.5)$. This is a relatively small square including all of the Mandelbrot set. The value of $l$ is initialized in the range $0.0001 \leq l \leq 0.01$ using a distribution whose natural log is uniformly distributed. Evolution is continued for 100,000 mating events, and the most
Fig. 3. The 20 best-of-run views of the Mandelbrot set found using a fitness function incorporating the second mask (constant function).

The model of evolution used is steady-state single tournament selection of size seven. Evolution proceeds by mating events in which seven views are selected. The two most fit are copied over the two least fit. The \( y \)-values of these copies are exchanged, and the \( l \)-values are exchanged 50% of the time. After the exchange of \( y \)-values and \( l \)-values the new structures are mutated. Mutation has two parts. The first consists of adding a displacement in the plane to \((x, y)\) in a direction selected uniformly at random and with magnitude given by a normal random variable with mean zero and standard deviation of 0.002. In the small range of zooms used in this study this constant value was acceptable, a greater range of potential zoom would require that this parameter be made self-adaptive to the current level of zoom. After the corner of the view has been displaced in this fashion, the side length is multiplied by \( e^{\ln(1.1) N(0,1)} \) where \( N(0,1) \) is a standard normal random variable. The effect of this is to multiply \( l \) by a value near 1. These mutation operators were chosen to permit incremental exploration of the view space with as little mutational bias as possible; separate mutation of the \( x \) and \( y \) coordinates, for example, would have biased search along the coordinate axes. The multiplicative mutation operator used for \( l \) makes small adjustments whose scale does not depend on the current value of the side length.

V. RESULTS AND DISCUSSION

Depictions of the 20 best-of-run views for the first mask are shown in Figure 2, for the second mask in Figure 3, and for the third mask in Figure 4.

Examining Figures 2-4 it is clear that changing the mask changes the character of view located. A minibrot is a smaller copy of the Mandelbrot set located somewhere within the Mandelbrot set. The first mask proved highly effective at locating minibrots with 14 of 20 views containing a minibrot. This is a reasonable outcome as a minibrot in the center of a view yields good agreement with the central 200 values in the mask.

The third mask located, in many cases, tendrils of the Mandelbrot set that track the ridge within the mask fairly well. Such minibrots as appear in these views are smaller and less central. The second mask, initially conceived as a control, produced the greatest diversity of views. Optimizing for the
sample points to have a divergence behavior of 150 seems to specify no particular character within the Mandelbrot set, but it does require some action. This mask functions as a “go find something” fitness function. It is also interesting to note that visible minibrots are far less apparent for the second mask.

All three masks checked yielded a substantial diversity of outcomes. While the masks permit some control of the type of view located, they do not unduly limit the variety of depictions that arise under their supervision.

One issue in any evolutionary computation problem is the nature of its fitness landscape. In each of the sets of 20 runs performed 20 different views were located. This suggests that the fitness landscapes for the fitness functions defined here are quite rough. Figure 5 shows the locations, not to scale, of the views located with the first mask. The views show preferences for certain parts of the Mandelbrot set, clustering to some degree.

VI. Generalizations

This section deals with two topics. The first is alternate fitness functions for searching the Mandelbrot set; the other is that of adapting the technique to search other fractals.

A. Other Fitness Functions for the Mandelbrot set

The fitness functions used here were not the first tried by the author for evolving interesting views in the Mandelbrot set. There are two regions that are good to avoid: views purely in the interior of the Mandelbrot set and views well outside it. Both these regions are boring, having little or no variation in divergence behavior or shape. The first fitness function the author tried maximized the sum of divergence behaviors less than \( M_{\text{max}} \) on the sample grid. This function attempts to place the sample points in locations with the largest possible divergence behavior while avoiding placing them on points in the Mandelbrot set at the resolution of sampling. The views located with this fitness function are striking but quite similar to one another. An example of one of these views is shown in Figure 6. The view depicted in Figure 6 is far denser than any of the views located using the mask-based fitness functions.

The following is a list of possible variations of the fitness function for searching the Mandelbrot set.

1) **Other mask functions.** The most obvious variation is to simply use other functions to generate the mask. The number of sample points and \( M_{\text{max}} \) could also be varied.
TABLE I
VALUES USED FOR THE FIRST AND THIRD FITNESS FUNCTION MASKS. THE SECOND FITNESS FUNCTION MASK TAKES ON A CONSTANT VALUE OF 150.

Fig. 6. A view located with an alternate fitness function that failed to find a diversity of appearances.

2) **Masks with indifference.** A special value could be included in masks that yields indifference to the value at the mask points having that value. This is a simple modification of the square mask to permit subsets of the mask to be active in fitness evaluation.

3) **Unstructured masks.** A mask could be given as a sequence \((X, Y, Value)\) at displacements \((\Delta X, \Delta Y)\) from the corner of the mask \((x, y)\) with \(Value\) being the target number of iterations at that point. This sort of unstructured mask would permit a far broader variety of searches of the Mandelbrot set.

4) **Induced masks.** A vast amount of effort has been expended by artists and hobbyists to locate interesting subsets of the Mandelbrot set. Taking a view already thought to be interesting and then building a mask from the divergence behavior at sample points within that view would yield a mask for which the view is already highly fit. This mask could in turn be used to evolve other views which *might* share the characteristics that made the initial view interesting. Induced masks yield another point at which the hand of the artist could be used to shape the evolutionary search for interesting views.

These ideas are offered as future directions for searching the Mandelbrot set; the author welcomes suggestions of other methods as well and notes that the entire issue of coloring the depictions has been avoided. Coloring is a rich field for investigation but not an especially rewarding one in a black-and-white publication.

**B. Other fractals**

The Mandelbrot set is more properly called the *quadratic* Mandelbrot set. It is based on iterated squaring. If \(f(z)\) is any complex-valued function of the complex numbers, then there is an associated Mandelbrot set based on the sequence

\[
\begin{align*}
  z_0 &= z \\
  z_{n+1} &= f(z_n) + z.
\end{align*}
\]

The cubic, quartic, quintic, and higher-order Mandelbrot sets have an appearance different from but reminiscent of the quadratic Mandelbrot set. The Mandelbrot sets based on transcendental functions such as sine and cosine have radically...
different appearances. All of these can be swapped into the current evolutionary algorithm with a minimum of effort and represent alternative sources of evolved art.

The Julia set is a fractal closely related to the Mandelbrot set. Rather than adding the point \( z \) under consideration when checking for set membership, a fixed complex constant \( w \) is added. The series used to define the Julia set is based on the sequence:

\[
\begin{align*}
    z_0 &= z \\
    z_{n+1} &= z_n^2 + w.
\end{align*}
\]

Julia sets are more repetitive at smaller scales than the Mandelbrot set, and so searching for values of \( w \) that yield interesting overall appearances is a more natural target for an evolutionary algorithm. This, in turn, suggests that unstructured masks would be a better choice than grid-shaped masks for Julia-set search. In addition, the Mandelbrot set indexes the Julia set in the following sense. The appearance of the Mandelbrot set near the value \( w \) is strongly predictive of the overall appearance of the Julia set [8]. Figure 7 shows some examples of this phenomenon. With such a ready visual index, the value of an evolutionary algorithm to search for interesting Julia sets is doubtful.

It was discovered, as the result of a bug in the fractal code under development, that the Julia set can be generalized in the following fashion. Let \( w_1 \) and \( w_2 \) be complex constants. Then the series:

\[
\begin{align*}
    z_0 &= z \\
    z_{2n+1} &= f(z_{2n}) + w_1 \\
    z_{2n} &= f(s_{2n-1}) + w_2
\end{align*}
\]

defines a new type of Julia set. If \( w_1 = w_2 \), then the result is a classic Julia set. When \( w_1 \) and \( w_2 \) are distinct, the resulting fractals exhibit properties, such as disconnected regions with positive area, that Julia sets are known not to exhibit. While a form of 4-dimensional Mandelbrot set does index these Julia sets, it is not viewable in any reasonable fashion. This suggests that a four-real-parameter evolutionary algorithm might be valuable for locating such generalized Julia sets. The generalization given is for two constants: there is an obvious generalization to any number of constants \( w_1, w_2, \ldots, w_k \) which yields ever more challenging targets for evolutionary search. Examples of such four-parameter generalized Julia sets are shown in Figure 8.

The entire discussion for Mandelbrot sets based on more general functions than iterated squaring applies to Julia sets as well, and the Mandelbrot set for a given complex function \( f(z) \) indexes the Julia sets for that function as it does for the quadratic case.

VII. ACKNOWLEDGMENTS

The first author would like to thank the University of Guelph Department of Mathematics and Statistics for its support of

Fig. 8. Examples of generalized Julia sets with four parameters (two complex constants).
Fig. 7. Indexing of Julia sets by the Mandelbrot set. Dots show the position of the constant $w$ used to generate the Julia sets shown as thumbnails.

References


