A NEW APPROACH TO IFS BOUNDING

ZALÁN BODÓ AND ANNA SOÓS

Babeş-Bolyai University, Department of Mathematics and Computer Science, Cluj-Napoca

Abstract The expansion of the attractor of a fractal described by an iterated function system can not be calculated, estimated by analytic methods. There is an analytic method to calculate the center and the radius of a bounding circle, but this circle is not optimal. Other researchers tried to find an optimal convex bounding polygon (for 2D IFS attractors), that is to have minimal area. I will present their method and I will propose a new approach for solving the bounding problem using genetic algorithms.

1 Introduction

Fractals are recursively defined "complex" structures. In this paper I will deal only with fractals described by iterated function systems (IFS) defined in the two-dimensional euclidean space. Fractal functions, iterated function systems are widely used in many domains of applied mathematics. For example fractal interpolation functions are used to approximate rough functions, curves in signal analysis and approximation, fractal image compression uses partitioned iterated function systems to describe a picture (which is supposed to be self-similar), etc.
In order to understand why is not so simple the estimation of the fractals' expansion, a brief introduction about fractals follows.

Fractals are complex, complicated figures, which can not be described by classical geometry. Their interesting property is that their dimension (not the classical topological dimension, but for example the similarity dimension or the Hausdorff dimension) is a fraction. From the article *How long is the coast of Britain?* published by Benoit B. Mandelbrot, creator of the *fractal* denomination, in the *Science Magazine* in 1967 begins the still actual arborescent research in the domain of fractals. K. Falconer gives a set of conditions which must be fulfilled by a figure to be a fractal, to say that has fractal structure ([11], 64):

- has fine structure
- it is too irregular to describe with classical geometry
- it is self-similar, or just approximately or statistically
- its fractal dimension is greater than its topological dimension
- it is easy to define (e.g. recursively)

The elements, objects of the nature are easier to describe, they can be better approximated with fractals than with polyhedrons, polygons. The trees, clouds, porifers all shows fractal structures. These patterns are so complex, that they could not be described before.

Figure 1. Two iterations of the Koch-curve.

The well-known Koch-curve is built up as follows: a line segment with length K_0 is divided into three equal parts. On the middle segment we build an equilateral triangle and we delete its undermost side. This process is repeated with the required rotation for the resulting four line segments. Then for all the line segments obtained, we repeat the process again and again, to infinity. The image we obtain is called the Koch-curve, and the process or procedure we followed is called the *copying machine algorithm* ([4], 3-4).

The transformations have to be applied recursively to obtain the image of the fractal. We can

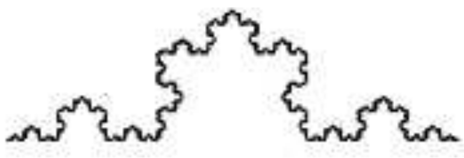

Figure 2. Obtained image of the Koch-curve after many iterations.

start the iteration also with a point, the image we obtain, called the attactor of the IFS, will be the same. That is, start with an arbitrary chosen initial point, apply all the transformations to that point, then for all the points obtained apply all the transformations, repeating this process to infinity. Evidently, infinity is not a "natural number", thus we settle for a big number. As we have seen, this process is recursive, furthermore we have to perform the operation *n*-times, where *n* is a big number, so the elimination of the recursion would be very favourable. Let us define first the iterated function systems.

Definition 1. *The* $f : R \to R,\ f(x) = ax + b,\ a,b \in R$ *function is called (one-dimensional) affine transformation which resizes with a and shifts with b.*

Definition 2. *The* $w : R \to R,\ w(x, y) = (ax + by + e,\ cx + dy + f),\ a, ..., f \in R$ *transformation is called a two-dimensional affine transformation. The following is an equivalent notation:*

$$w\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a\ b \\ c\ d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = A\begin{pmatrix} x \\ y \end{pmatrix} + T.$$

Definition 3. *Let* (X,d) *a complete metric space. The* $\{X; w_n, n = 1,\ldots,N\}$ *finite set of contraction mappings* $w_n : X \to X$*,* $n = 1,2,...,N$ *is called an iterated function system (IFS). If contractivity factor of* w_n *is* s_n*, then the contractivity factor of the IFS is* $s = \max\{s_n \mid n = 1,2,\ldots,N\}$.

For example the Koch curve's iterated function system is the following:

$$w_1\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.33 & 0\\0 & 0.33\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0\\0\end{pmatrix}$$
$$w_2\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.33 & 0\\0 & 0.33\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.66\\0\end{pmatrix}$$
$$w_3\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.16 & -0.28\\0.28 & 0.16\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.33\\0\end{pmatrix}$$
$$w_4\begin{pmatrix}x\\y\end{pmatrix} = \begin{pmatrix}0.16 & 0.28\\-0.28 & 0.16\end{pmatrix}\begin{pmatrix}x\\y\end{pmatrix} + \begin{pmatrix}0.49\\0.28\end{pmatrix}$$

Barnsley's *Chaos Game* algorithm eliminates the recursion, and the attractor will be the same as we had used the recursive method.
The *Chaos Game* algorithm:

> Let $\{X; w_n, n = 1,\ldots,N\}$ an IFS.
> k:=0
> Choose a starting point (x_0, y_0) and trace it out.
> While *terminal condition* do:
>   Choose a random integer, such that $r \in \{1,\ldots N\}$
> $(x_{k+1}, y_{k+1}) := w_r(x_k, y_k)$
> $k := k+1$
>   Trace out (x_k, y_k)
> End of While

The proof of the working of the above algorithm has two steps. First must be proved that independently of the starting (initial) point, the obtained image converges to the attractor of the IFS. In the second step it needs to be proved that if a point obtained by the iteration of the algorithm is on the attractor, all the produced points hereafter will be on the attractor.

Another version of the algorithm works with probabilities assigned to each member function. The probability p_i is calculated from the matrix A of the transformations:

$$p_i = \frac{|detA_i|}{\sum_{j=1}^{N}|detA_j|}, \quad i = 1,\ldots,N \text{ (if } detA_i = 0 \text{ then } p_i \text{ will be a small number, e.g. 0.001.)}$$

In order to be able to apply the transformations according to these probabilities, the q_i cumulative probabilities of the transformations must be calculated. Generating a random number r between 0 and 1, we choose the transformation to be applied as follows: if $r \leq q_1$ then we choose the first transformation; if $q_i < r \leq q_{i+1}, i = 1, \ldots, N-1$ then we choose the $(i+1)^{\text{th}}$ one.

We could see, that the elimination of the recursion from the fractal drawing algorithm does not eliminate the recursion present in the fractal's structure. Because of its recursive structure we can not decide if a point lies on the attractor or not. Furthermore for a given x we can not calculate y. That is why attractor expansion estimation is problematic.

2 IFS polygon bounding

When a fractal needs to be drawn, it is an unknown fact where the attractor will lie (because of its recursive structure). Usually for determination of its expansion the bounding circle algorithm is used ([5], [9]). This provides a non-optimal circle, in which the attractor surely lies. An interesting problem in computer graphics is to calculate the ray-fractal intersection of 3D IFS fractals. Lawlor and Hart ([7], [6]) used the *recursive bounding theorem* to determine such a bounding hull. Then the fractal can be raytraced based on its bounding polyhedron. The theorem states that if a bounding volume contains its images under the maps of the IFS, then it contains the attractor of the IFS. In this section I present the method used in [7] and [6]. Lawlor and Hart used a linear optimizer to solve the problem; the objective function being optimized (minimized) was the sum of the displacements, although it can also be the area of the polygon or the sum of the distances from one vertex of the polygon to the others, and the constraints were similar to those which will be presented.

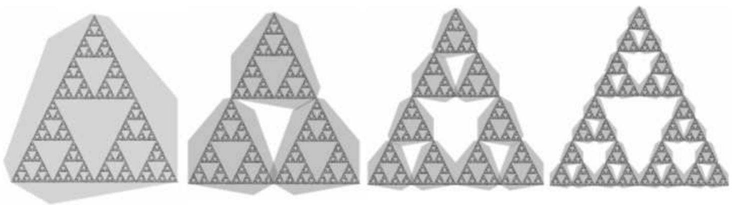

Figure 3. Figure from [7] demonstrating the statement of the theorem.

Theorem 1 (Recursive Bounding Theorem). *Let B a non-empty compact subset of $\mathbf{R}^n$ and let W be the Hutchinson operator (the union of the maps of the IFS) of a convergent IFS on $\mathbf{R}^n$. If $W(B) \subset B$ then $W^\infty(B) \subset B$.*

Proof. The theorem is proved using induction.
Let $W(B) \subset B$. For some k suppose that $W^k(B) \subset B$. Let introduce the notation $C = W^k(B)$. $C \subset B \Leftrightarrow w_m(C) \subset w_m(B), \forall m \in \underbrace{\{1, \ldots, N\}}_{M} \Rightarrow W(C) = \bigcup_{m \in M} w_m(C) \subset \bigcup_{m \in M} w_m(B) = W(B)$.

Then $W(W^k(C)) = W^{k+1}(C) \subset W(B) \subset B$. By induction, we have $W^\infty(B) \subset B$. ([7]) ■

Our goal is to find an optimal (convex) polygon, such that to contain the transformed polygons by the maps of the IFS.

2.1 Lawlor and Hart's algorithm

The normals of the lines constituting the polygon are fixed, Lawlor and Hart picked equally spaced directions, which means that they picked equally spaced points on a circle, from where they calculated the normals of the lines.

The normal of a line through points (x_1, y_1) and (x_2, y_2) is $\overrightarrow{n} = (y_1 - y_2 \quad x_2 - x_1)$, and the displacement is $x_1y_2 - x_2y_1$. That is the equation of the line is

$$\begin{pmatrix} y_1 - y_2 & x_2 - x_1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + x_1y_2 - x_2y_1 = 0$$

A point (x_0, y_0) lies inside the polygon, if

$$\overrightarrow{n_i} \cdot \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + d_i \leq 0, \forall i = 1, \ldots, nr\_points,$$

where $\overrightarrow{n_i}$ is the normal, and d_i is the displacement of the i^{th} line constituting the polygon. According to the *recursive bounding theorem*, the transformed polygons by the member functions of the IFS lies is the interior of the original (bounding) polygon if

$$\overrightarrow{n_i} \cdot w_j \begin{pmatrix} x_k \\ y_k \end{pmatrix} + d_i \leq 0, \forall i \in I, j \in J, k \in K.$$

The above formula means that we have to check for all the vertices of the bounding polygon, for all the transformations, and for all the polygon sides (lines) if the formula gives a value less or equal then zero.
The intersection of two lines can be calculated from the formula given below

$$\overrightarrow{x_{ij}} = -N_{ij}^{-1} \cdot \begin{pmatrix} d_i \\ d_j \end{pmatrix},$$

where

$$N_{ij} = \begin{pmatrix} \overrightarrow{n_i} \\ \overrightarrow{n_j} \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ a_j & b_j \end{pmatrix}, \text{ therefore } N_{ij}^{-1} = \frac{1}{a_ib_j - a_jb_i} \cdot \begin{pmatrix} b_j & -b_i \\ -a_j & a_i \end{pmatrix}$$

3 A new approach to IFS polygon bounding

The base of this approach is also the *recursive bounding theorem*. We seek for a minimal area bounding polygon, such that the transformed polygons by the maps of the IFS to be in the interior of the bounding polygon. The optimization algorithm is a genetic algorithm.

The basic idea consists in the fact that a convex polygon transformed by an affine transformation remains also convex. The genetic algorithm will search for such an affine transformation (6 parameters) that a previously fixed polygon transformed by this function to have minimal area and fulfill the required conditions.

The base polygon (which remains fixed) is a set of *nr_points* equally spaced points on the bounding circle, determined by the following algorithm.

3.1 Finding a non-optimal bounding circle

In order to fill the initial population with legal data, and to be able to set the lower and upper bounds of the variables, it is useful to determine a non-optimal bounding circle for the attractor. In [5] Hart and DeFanti give a convergent iterative algorithm for determining a bounding circle for an IFS. Starting from an initial point (e.g. (0,0)) as the center and with an initial radius (of length 1), the new center and radius is given by the following two formulae:

$$x_{i+1} = \frac{1}{N} \cdot \sum_{j=1}^{N} w_j(x_i), \quad r_{i+1} = \max_{j=1,\dots,N} \{d(w_j(x_i), x_{i+1}) + s_j \cdot r_i\}$$

The obtained bounding circle is $C(x_\infty, r_\infty)$, the approximation of it can be calculated with a larger number of iterations. In the second formula s_j is the contractivity factor of the j^{th} affine transformation, which can be calculated from the next formula ([3]):

$$s_i = \sqrt{\alpha + \beta + \sqrt{(\alpha - \beta)^2 + \gamma^2}}$$

$$\alpha = \frac{a_i^2 + c_i^2}{2}, \quad \beta = \frac{b_i^2 + d_i^2}{2}, \quad \gamma = a_i b_i + c_i d_i$$

In [3], [10] we can find another method which finds the same bounding circle for an IFS; their technique is an analytic one, so requires not so many iterations, but computationally is much more complicated. Mixing the methods in [9] and [10] we can determine the required values analytically, by the following procedure. The transformations of an IFS are of the form

$$w_i(x) = A_i x + B_i, \quad i = 1, \dots, N, x \in \mathbf{R}^2.$$

Let us define

$$A = \frac{1}{N} \sum_{i=1}^{N} A_i$$

and

$$B = \frac{1}{N}\sum_{i=1}^{N} B_i.$$

Then x_∞ is the fixed point of the affine function $f(x) = Ax + B$, which can be easily calculated:

$$x_\infty = (I - A)^{-1}B.$$

The radius of the bounding circle can be calculated as follows:

$$r_\infty = \max_{i=1}^{N} \frac{d(x_\infty, w_i(x_\infty))}{1 - s_i}.$$

3.2 Lower and upper bounds determination

We have to determine an interval for each of the 6 components of an individual. We consider the following 2D affine transformations: scale/stretch, skew in the x-direction, skew in the y-direction, rotation, displace. All of these transformations can be represented by a 2×2 coefficient matrix except the last one, which is a column vector.

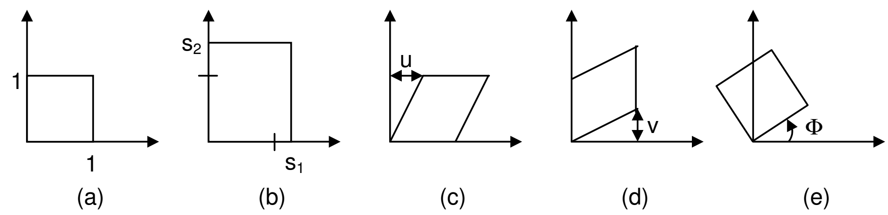

Figure 4. Illustration of the four type of affine transformations: (a) original square, (b) scale/strech, (c) skew in x-direction, (d) skew in y-direction, (e) rotation.

The coefficient matrices for the above transformations are the following:

$$A_1 = \begin{pmatrix} s_1 & 0 \\ 0 & s_2 \end{pmatrix} \quad A_2 = \begin{pmatrix} 1 & u \\ 0 & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & 0 \\ v & 1 \end{pmatrix} \quad A_4 = \begin{pmatrix} \cos\Phi & -\sin\Phi \\ \sin\Phi & \cos\Phi \end{pmatrix}$$

We are searching for a v 2D affine transformation with the following form

$$v\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

We can set the components of the offset vector to vary in the $[-r, r]$ interval, but this interval can be greater. Because the polygon can not be greater than the initial one, $s_1, s_2 \in [0, 1]$. (This statement is not true if we deal with a small number of polygon vertices.) Evidently, $\Phi \in [0, 2\pi)$. I also set the u, v skew parameters to vary in the $[-r, r]$ interval. Then, multiplying the coefficient matrices, we get the following intervals for the parameters: $a, b \in [-r^2 - r - 1, r^2 + r + 1]$, $c, d \in [-r - 1, r + 1]$, and for the offsets $e, f \in [-r, r]$.

3.3 Fitness function, constraints handling and genetic operators

In order to get an optimal convex bounding polygon for an IFS, we have to minimize the area of the polygon according to the constraints, that is the transformed polygons to lie in the interior of the bounding polygon. Since we search for the maxima of the fitness function, the area, or a similar measure (like the sum of the distances from one vertex to the others, which will be denoted by SD), which should give results similar to minimizing the area, will be in the denominator of the fitness function. Thus

$$f(Ind) = \frac{10^l}{SD_{polygon}(Ind) + t \cdot penalty(Ind)},$$

where Ind is the individual, $SD_{polygon}$ is the sum of the distances from an arbitrary chosen vertex to the others in the polygon, and t is the actual generation (iteration) number. The penalty-function penalizes those situations when the transformed polygons do not lie in the interior of the bounding polygon.

The transformation we are searching for, has two components, thus the v can be used with one argument; then we apply the corresponding component of the 2D map. The penalty function will be

$$penalty(Ind) = \sum \max\left(0, \left(v(y_i) - v(y_{i+1})\ v(x_{i+1}) - v(x_i)\right) \cdot w_j \left(v \begin{pmatrix} x_k \\ y_k \end{pmatrix}\right) + \right.$$
$$\left. v(x_i) \cdot v(y_{i+1}) - v(x_{i+1}) \cdot v(y_i)\right) \quad i = 1, \ldots, nr\_points, j = 1, \ldots, N, k = 1, \ldots, nr\_points$$

where N is the number of the maps of the IFS, $(x_i, y_i), i = 1, \ldots, nr\_points + 1$ are the vertices of the polygon on the bounding circle, $(x_{nr\_points+1}, y_{nr\_points+1}) = (x_1, y_1)$.

For crossing over a convex crossing over operator was used, which generates two offsprings, and for mutation a non-uniform mutation; these operators are used by the GENOCOP system for solving general nonlinear problems in convex search spaces ([8]). If an individual or chromosome contains q genes then we use the following operators:

- crossing over: $x_1, x_2 \to x'_1, x'_2$

$$\begin{aligned} x'_1 &= (x_1, \ldots, x_k, y_{k+1}a + x_{k+1}(1-a), \ldots, y_q a + x_q(1-a)) \\ x'_2 &= (y_1, \ldots, y_k, x_{k+1}a + y_{k+1}(1-a), \ldots, x_q a + y_q(1-a)), \end{aligned}$$

where $a \in [0, 1]$.

- non-uniform mutation:

$$x'_k = \begin{cases} x_k + \Delta(t, upper\_bound(k) - x_k), & \text{if a random binary digit is 0} \\ x_k - \Delta(t, x_k - lower\_bound(k)), & \text{if a random binary digit is 1} \end{cases}, k \in [1, q]$$

where

$$\Delta(t,y) = y \cdot r \cdot (1 - \frac{t}{T})^b.$$

The t parameter is the number of the actual generation, T is the number of the overall generations being created, and b is a number which determines the degree of non-uniformity. The non-uniformity of this operator consists in getting closer to the end of the iterations, the mutation performs in a smaller and smaller interval.

3.4 Discussion and results

Fractal	Points' number	Population size	Generations	p_c	p_m	SD	Bounding
Barnsley's fern	8	30	1000	0.5	0.4	48.382	yes
Barnsley's fern	10	30	1000	0.5	0.4	63.383	yes
Branch	8	30	1000	0.5	0.4	75.831	yes
Branch	10	30	1000	0.5	0.4	86.906	yes
Golden dragon	8	30	1000	0.5	0.4	57.431	not really
Golden dragon	10	30	1000	0.5	0.4	72.504	not really

For the second experiment the polygon points on the bounding circle, the minimal *SD* and the best individual (best transformation) after 1000 generations, with a population of size 30, and with $p_c = 0.5$, $p_m = 0.4$ were the following:

- polygon points: (7.536,6.899), (2.896,10.270), (-2.840,10.270), (-7.480,6.899), (-9.253,1.444), (-7.480,-4.011), (-2.840,-7.383), (2.896,-7.383), (7.536,-4.011), (9.309,1.444)

- *SD*: 63.383

- transformation: $v\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 0 & 0.364 \\ 0.596 & 0.164 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0.007 \\ 4.929 \end{pmatrix}$

The algorithm finding an optimal transformation does not provide such a good minimal bounding polygon as the other algorithm, which is evident, because this method can not vary the displacements of the polygon lines, to enclose better and better the attractor. Although the genetic algorithm converges very fast to the optimal polygon, to the optimal transformation.

We can observe that the probability of mutation in the experiments is rather large. This is because of the numerosity of local optimas of the fitness function.

By the new method the running time of the optimization algorithm is not influenced by the number of the polygon vertices, because the number of the parameters of an affine transformation remains unchanged.

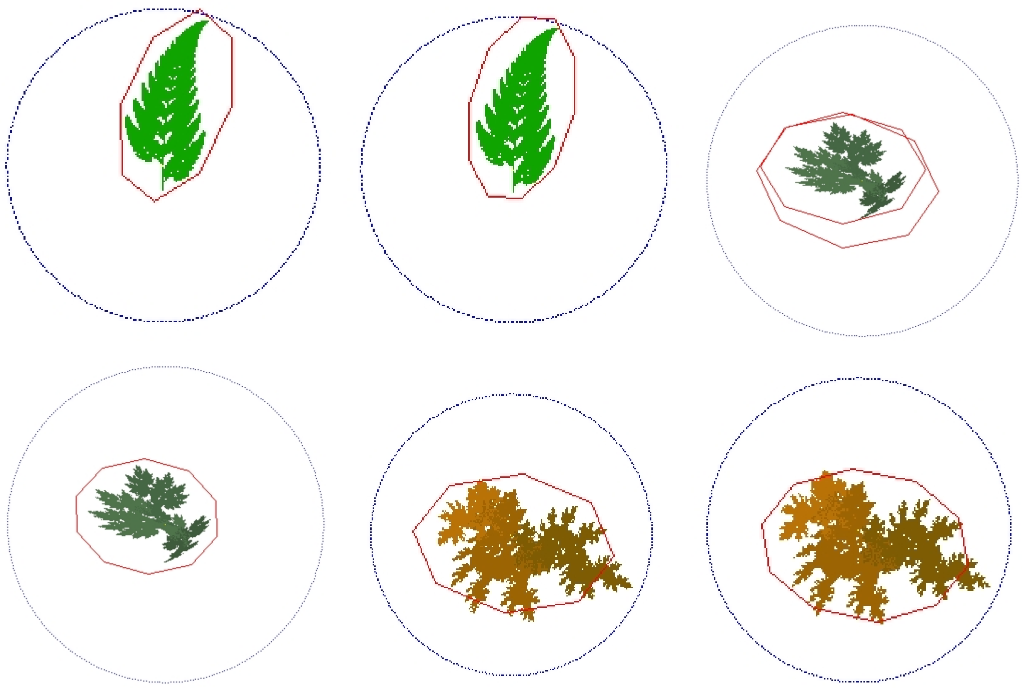

Figure 5. The optimal polygons found by the algorithm.

An interesting fact is that the results are influenced by l, power of 10 in the counter of the fitness function. It might be 0, but then all values would be in the $[0,1]$ interval, which encumbers the work of the algorithm, because of the loss of accuracy, resulting from the floating point representation of the numbers. The best results were obtained for $l = 4$.

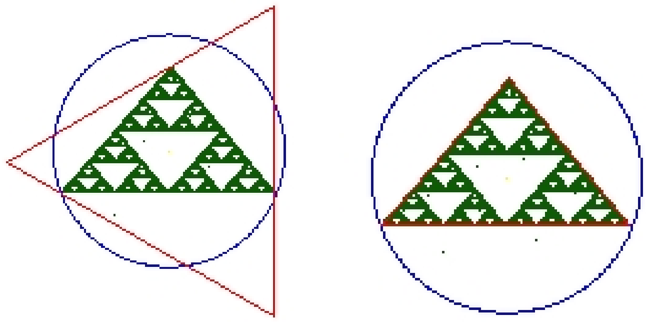

Figure 6. Results obtained for the Sierpinski triangle with the first and the second algorithm respectively; number of points=3, population size=30, generations number=1000, $l = 4$, $p_c = 0.5$, $p_m = 0.4$.

References

[1] BARNSLEY, MICHAEL F., *Fractals Everywhere*, second edition, Morgan Kaufmann, 1993.

[2] BARNSLEY, MICHAEL F., Fractal Functions and Interpolation *Constructive Approximation*, Springer-Verlag, New York, 1986.

[3] EDALAT, A., SHARP, D.W.N., WHILE, R.L., *Bounding the Attractor of an IFS*, Imperial College Research Report DoC 96/5, 1999. (www.doc.ic.ac.uk/research/technicalreports/1996/DTR96-5.pdf)

[4] FISHER, YUVAL (editor), *Fractal Image Compression - Theory and Application*, Springer-Verlag, New York, 1996.

[5] HART, J.C., DEFANTI, T.A., Efficient Antialiased Rendering of 3-D Linear Fractals, *Computer Graphics*, Volume 25, Number 4, July 1991. (http://graphics.cs.uiuc.edu/ jch/papers/rayifs.pdf)

[6] LAWLOR, ORION SKY, *Bounding Iterated Function Systems via Convex Optimization*, Department of Computer Science, University of Illinois at Urbana-Champaign.

[7] LAWLOR, O.S., HART, J.C., *Bounding Recursive Procedural Models using Convex Optimization*, Department of Computer Science, University of Illinois at Urbana-Champaign. (http://graphics.cs.uiuc.edu/∼jch/papers/ifsbvopt.pdf)

[8] MICHALEWICZ, ZBIGNIEW, *Genetic Algorithms+Data Structures=Evolution Programs*, Springer-Verlag, 1996.

[9] RICE, JONATHAN, *Spatial Bounding of Self-Affine Iterated Function System Attractor Sets*, Image Synthesis Group, Department of Computer Science, Trinity College, Dublin, Ireland. (www.graphicsinterface.org/proceedings/1996/Rice/rice.ps.gz)

[10] SHARP, D.W.N., WHILE, R.L., *A Tighter Bound on the Area Occupied by a Fractal Image*, Imperial College Research Report DoC 96/5, 1999. (www.doc.ic.ac.uk/research/technicalreports/1996/DTR96-6.pdf)

[11] SZABÓ LÁSZLÓ IMRE, *Ismerkedés a fraktálok matematikájával*, Polygon, Szeged, 1997.