Memetic Modified Cuckoo Search Algorithm with ASSRS for the SSCF Problem in Self-Similar Fractal Image Reconstruction

Akemi Gálvez[1,2], Andrés Iglesias[1,2,†], Iztok Fister[3], Iztok Fister Jr.[3], Eneko Osaba[4], Javier Del Ser[4,5,6]

[1]Toho University, 2-2-1 Miyama, 274-8510, Funabashi, Japan
[2]University of Cantabria, Avenida de los Castros s/n, 39005, Santander, Spain
[3]University of Maribor, Smetanova, Maribor, Slovenia
[4]TECNALIA, Derio, Spain
[5]University of the Basque Country (UPV/EHU), Bilbao, Spain
[6]Basque Center for Applied Mathematics (BCAM), Bilbao, Spain
[†]Corresponding Author: `iglesias@unican.es`
`http://personales.unican.es/iglesias`

Abstract. This paper proposes a new memetic approach to address the problem of obtaining the optimal set of individual *Self-Similar Contractive Functions* (SSCF) for the reconstruction of self-similar binary IFS fractal images, the so-called SSCF problem. This memetic approach is based on the hybridization of the modified cuckoo search method for global optimization with a new strategy for the Lévy flight step size (MMCS) and the adaptive step size random search (ASSRS) heuristics for local search. This new method is applied to some illustrative examples of self-similar fractal images with satisfactory graphical and numerical results. Our approach represents a substantial improvement with respect to a previous method based on the original cuckoo search algorithm for all contractive functions of the examples in this paper.

Keywords: image reconstruction, swarm intelligence, cuckoo search algorithm, fractal images, iterated function systems, contractive functions

1 Introduction

Fractals are one of the most interesting mathematical objects ever defined. They are also very popular in science due to their ability to describe many growing patterns and natural structures (branches of trees, river networks, coastlines, mountain ranges, and so on). Furthermore, fractals have also found remarkable applications in computer graphics, scientific visualization, image processing, dynamical systems, medicine, biology, arts, and other fields [1, 2, 8–10].

One of the most popular methods to obtain fractals images is the *Iterated Function Systems* (IFS), conceived by J.E. Hutchinson [11] and popularized by M. Barnsley in [1]. Roughly, an IFS consists of a finite system of contractive maps on a complete metric space. Any IFS has a unique non-empty compact fixed set $\mathcal{A}$ called the *attractor of the IFS*. The graphical representation of this attractor is (at least approximately) a self-similar fractal image. Conversely, each self-similar fractal image can be represented by an IFS. Obtaining the parameters of such IFS is called the *IFS inverse problem*. Basically, it consists of solving an image reconstruction problem: given a self-similar fractal image, determine the IFS whose attractor approximates such input image accurately. This IFS inverse problem is so difficult that only partial solutions have been reached so far. A very promising strategy is to split up the problem into two steps: firstly, obtain a suitable collection of individual self-similar contractive functions for the IFS, the so-called *SSCF problem*. The output of this step is then applied to compute the optimal solution for the general IFS inverse problem.

A previous paper addressed this first step by using the cuckoo search (CS) algorithm [13]. Although the method provided nice visual results, its accuracy was far from optimal, and can still be improved. Recently, the original CS has been improved and modified for better performance. In this sense, the present paper proposes a new hybrid scheme based on the CS and called *Memetic Modified Cuckoo Search* (MMCS). Our approach combines two techniques: firstly, we consider a variant proposed in [17] of the original cuckoo search algorithm for global optimization and called *Modified Cuckoo Search* (MCS). This variant is based on two important modifications: (1) the value of the Lévy flight step size is changed dynamically with the iterations; (2) the addition of information exchange between the eggs to speed up convergence to the optimum. In our approach, the Lévy flight step size is changed according to a new strategy proposed in this paper. This technique is hybridized with the *Adaptive Step Size Random Search* (ASSRS), a local search heuristics based on changing adaptively the radius of the hypersphere around the most promising solutions for higher accuracy and to escape from local optima.

The structure of this paper is as follows: Section 2 introduces the mathematical background about the iterated function systems and the SSCF problem. Then, Section 3 describes the original and the modified cuckoo search algorithms. Our proposed MMCS method is described in detail in Section 4, while the experimental results are briefly discussed in Section 5. The paper closes with the main conclusions and some ideas about future work in the field.

2 Mathematical Background

2.1 Iterated Function Systems

An *Iterated Function System* (IFS) is a finite set $\{\phi_i\}_{i=1,\ldots,\eta}$ of contractive maps $\phi_i : \mathbf{\Omega} \longrightarrow \mathbf{\Omega}$ defined on a complete metric space $\mathcal{M} = (\mathbf{\Omega}, \mathbf{\Psi})$, where $\mathbf{\Omega} \subset \mathbb{R}^n$ and $\mathbf{\Psi}$ is a distance on $\mathbf{\Omega}$. We refer to the IFS as $\mathcal{W} = \{\mathbf{\Omega}; \phi_1, \ldots, \phi_\eta\}$. For visualization purposes, in this paper we consider that the metric space $(\mathbf{\Omega}, \mathbf{\Psi})$ is $\mathbb{R}^2$ along with the Euclidean distance d_2, which is a complete metric space. In this case, the affine transformations ϕ_κ are of the form:

$$\begin{bmatrix} \xi_1^* \\ \xi_2^* \end{bmatrix} = \phi_\kappa \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \begin{bmatrix} \theta_{11}^\kappa & \theta_{12}^\kappa \\ \theta_{21}^\kappa & \theta_{22}^\kappa \end{bmatrix} . \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} + \begin{bmatrix} \sigma_1^\kappa \\ \sigma_2^\kappa \end{bmatrix} \tag{1}$$

or equivalently: $\mathbf{\Phi}_\kappa(\mathbf{\Xi}) = \mathbf{\Theta}_\kappa.\mathbf{\Xi} + \mathbf{\Sigma}_\kappa$ where $\mathbf{\Sigma}_\kappa$ is a translation vector and $\mathbf{\Theta}_\kappa$ is a 2×2 matrix with eigenvalues $\lambda_1^\kappa, \lambda_2^\kappa$ such that $|\lambda_j^\kappa| < 1$. In fact, $\mu_\kappa = |det(\mathbf{\Theta}_\kappa)| < 1$ meaning that ϕ_κ shrinks distances between points. Let us now define a transformation, Υ, in the set of compact subsets of $\mathbf{\Omega}$, $\mathcal{H}(\mathbf{\Omega})$, by

$$\Upsilon(\mathcal{S}) = \bigcup_{\kappa=1}^{\eta} \phi_\kappa(\mathcal{S}). \tag{2}$$

If all the ϕ_κ are contractions, Υ is also a contraction in $\mathcal{H}(\mathbf{\Omega})$ with the induced Hausdorff metric [1, 11]. Then, according to the fixed point theorem, Υ has a unique fixed point, $\Upsilon(\mathcal{A}) = \mathcal{A}$, called the *attractor of the IFS*.

Let us now consider a set of probabilities $\mathcal{P} = \{\omega_1, \ldots, \omega_\eta\}$, with $\sum_{\kappa=1}^{\eta} \omega_\kappa = 1$. There exists an efficient method, known as *probabilistic algorithm*, for the generation of the attractor of an IFS. Picking an initial point ξ_0, one of the mappings in the set $\{\phi_1, \ldots, \phi_\eta\}$ is chosen at random using the weights $\{\omega_1, \ldots, \omega_\eta\}$ and then applied to generate a new point; the same process is repeated again with the new point and so on. As a result, we obtain a sequence of points that converges to the fractal as the number of points increases. This set of points represents graphically the attractor of the IFS.

2.2 The Self-Similar Contractive Functions (SSCF) Problem

Suppose that we are given an initial self-similar fractal image $\mathcal{I}^\square$. The *Collage Theorem* says that it is possible to obtain an IFS $\mathcal{W}$ whose attractor has a graphical representation $\mathcal{I}^\blacksquare$ that approximates $\mathcal{I}^\square$ accurately according to a error function $\mathcal{E}$ between $\mathcal{I}^\square$ and $\mathcal{I}^\blacksquare$. Note that $\mathcal{I}^\blacksquare = \Upsilon(O^\square)$ for any image $O^\square$. Mathematically, this means that we have to solve the optimization problem:

$$\underset{\{\mathbf{\Theta}_\kappa, \mathbf{\Sigma}_\kappa, \omega_\kappa\}_{\kappa=1,\ldots,\eta}}{minimize} \quad \mathcal{E}\left(\mathcal{I}^\square, \Upsilon(O^\square)\right) \tag{3}$$

which is a continuous constrained optimization problem, since all free variables in $\{\mathbf{\Theta}_\kappa, \mathbf{\Sigma}_\kappa, \omega_\kappa\}_\kappa$ are real-valued and must satisfy the condition that all ϕ_κ have to be contractive. It is also a multimodal problem, since there can be several global or local minima of the error function. So far only partial solutions have been reported, but the general problem remains unsolved.

A promising strategy to tackle this issue is to solve firstly the sub-problem of computing a suitable collection of self-similar contractive functions for the IFS (this is called the *SSCF problem*). However, even this SSCF problem is challenging because we do not have any information about the number of contractive functions and their parametric values. To overcome this limitation, a previous paper applied a given number of contractive maps ϕ_κ onto the original fractal image $\mathcal{I}^\square$ and compare the resulting images according to the error function $\mathcal{E}$ in order to obtain suitable values for the SSCF parameters [13]. With this strategy, the original problem (3) was transformed into the optimization problem:

$$\underset{\{\mathbf{\Theta}_\kappa, \mathbf{\Sigma}_\kappa, \omega_\kappa\}_{\kappa=1,\dots,\eta}}{minimize} \quad \mathcal{E}\left(\mathcal{I}^\square, \phi_\kappa(\mathcal{I}^\square)\right) \qquad\qquad (\kappa = 1, \dots, \eta) \tag{4}$$

The cuckoo search algorithm was applied to solve this optimization problem [13]. Unfortunately, although the reconstructed figures looked nice visually, the accuracy was far from optimal in terms of the numerical similarity error rates. In this paper, we modify that CS-based method to improve those results.

3 The Cuckoo Search Algorithms

3.1 Original Cuckoo Search (CS)

The *cuckoo search* (CS) is a powerful metaheuristic algorithm originally proposed by Yang and Deb in 2009 [19]. Since then, it has been successfully applied to difficult optimization problems [4, 12, 18, 20]. The algorithm is inspired by the obligate interspecific brood-parasitism of some cuckoo species that lay their eggs in the nests of host birds of other species to escape from the parental investment in raising their offspring and minimize the risk of egg loss to other species.

This interesting breeding behavioral pattern is the metaphor of the cuckoo search metaheuristic approach for solving optimization problems. In this algorithm, the eggs in the nest are interpreted as a pool of candidate solutions while the cuckoo egg represents a new coming solution. The ultimate goal of the method is to use these new (and potentially better) solutions associated with the parasitic cuckoo eggs to replace the current solution associated with the eggs in the nest. This replacement, carried out iteratively, will eventually lead to a very good solution of the problem. In addition to this representation scheme, the CS algorithm is also based on three idealized rules [19, 20]:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest;
2. The best nests with high quality of eggs (solutions) will be carried over to the next generations;
3. The number of available host nests is fixed, and a host can discover an alien egg with a probability $p_a \in [0, 1]$. For simplicity, this assumption can be approximated by a fraction p_a of the n nests being replaced by new nests (with new random solutions at new locations).

The basic steps of the CS algorithm are summarized in Table 1. It starts with an initial population of n host nests and it is performed iteratively. The initial values of the jth component of the ith nest are determined by the expression $x_i^j(0) = rand.(up_i^j - low_i^j) + low_i^j$, where up_i^j and low_i^j represent the upper and lower bounds of that jth component, respectively, and $rand$ represents a standard uniform random number on the interval $(0, 1)$. With this choice, the initial values are within the search space domain. These boundary conditions are also controlled in each iteration step. For each iteration t, a cuckoo egg i

Table 1. Cuckoo search algorithm via Lévy flights as originally proposed in [19, 20].

Algorithm: Cuckoo Search via Lévy Flights

begin
  Objective function $f(\mathbf{x})$, $\mathbf{x} = (x_1, \ldots, x_D)^T$
  Generate initial population of n host nests $\mathbf{x}_i$ $(i = 1, 2, \ldots, n)$
  while ($t < MaxGeneration$) or (stop criterion)
    Get a cuckoo (say, i) randomly by Lévy flights
    Evaluate its fitness F_i
    Choose a nest among n (say, j) randomly
    if $(F_i > F_j)$
      Replace j by the new solution
    end
    A fraction (p_a) of worse nests are abandoned and new ones are built via Lévy flights
    Keep the best solutions (or nests with quality solutions)
    Rank the solutions and find the current best
  end while
  Postprocess results and visualization
end

is selected randomly and new solutions $\mathbf{x}_i^{t+1}$ are generated by using the Lévy flight. The general equation for the Lévy flight is given by:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha \oplus levy(\lambda) \tag{5}$$

where $\alpha > 0$ indicates the step size (usually related to the scale of the problem) and $\oplus$ indicates the entry-wise multiplication. The second term of Eq. (5) is a transition probability modulated by the Lévy distribution as:

$$levy(\lambda) \sim t^{-\lambda}, \qquad (1 < \lambda \leqslant 3) \tag{6}$$

which has an infinite variance with an infinite mean. The authors in [20] suggested to use the Mantegna's algorithm, which computes the factor:

$$\hat{\phi} = \left(\frac{\Gamma(1+\hat{\beta}).sin\left(\frac{\pi.\hat{\beta}}{2}\right)}{\Gamma\left(\left(\frac{1+\hat{\beta}}{2}\right).\hat{\beta}.2^{\frac{\hat{\beta}-1}{2}}\right)} \right)^{\frac{1}{\hat{\beta}}} \tag{7}$$

where Γ denotes the Gamma function and $\hat{\beta} = 3/2$ in [20]. This factor is used in Mantegna's algorithm to compute the step length as: $\varsigma = u/|v|^{\frac{1}{\hat{\beta}}}$, where u and v follow the normal distribution of zero mean and deviation σ_u^2 and σ_v^2, respectively, where σ_u obeys the Lévy distribution given by Eq. (7) and $\sigma_v = 1$. Then, the stepsize ζ is computed as $\zeta = 0.01\,\varsigma\,(\mathbf{x} - \mathbf{x}_{best})$. Finally, $\mathbf{x}$ is modified as: $\mathbf{x} \leftarrow \mathbf{x}+\zeta.\mathbf{\Delta}$ where $\mathbf{\Delta}$ is a random vector that follows the normal distribution $N(0,1)$. The CS evaluates the fitness of the new solution and compares it with the current one. In case that the new solution brings better fitness, it replaces the current one. On the other hand, a fraction of the worse nests are abandoned and replaced by new solutions to increase the exploration of the search space looking for more promising solutions. The rate of replacement is given by the probability p_a, a parameter of the model that has to be tuned for better performance. Moreover, for each iteration step, all current solutions are ranked according to their fitness and the best solution reached so far is stored as the vector $\mathbf{x}_{best}$.

3.2 Modified Cuckoo Search (MCS)

The *modified cuckoo search* (MCS) method [17] aims at improving the performance of the original CS described above through two important modifications:

1. the value of the Lévy flight step size, α, assumed constant in the CS, is decreased with the number of iterations. The reason is to promote local search as the individuals get closer to the solution, in a rather similar way to the inertia weight in PSO. In [17] an initial value of the Lévy step size $\alpha^0 = 1$ is chosen. At each generation t, the new step size is computed adaptively as:
$$\alpha^t = \frac{\alpha^0}{\sqrt{t}} \tag{8}$$
This modification is only applied on the set of nests to be abandoned.
2. the addition of information exchange between the eggs to speed up convergence to the optimum. In the original CS, the search relies on random walks so fast convergence is not guaranteed. In the MCS, some eggs with the best fitness are selected for a set of top eggs. For each of the top eggs, a second egg is chosen randomly and then a third egg is generated on the path from the top egg to the second one, at a distance given by the inverse of the golden ratio $\varphi = (1+\sqrt{5})/2$, so that it gets closer to the top egg.

With these modifications, the MCS performs better than the CS for several examples, showing a higher convergence rate to the actual global minimum [17].

4 Proposed Approach

4.1 Memetic Modified Cuckoo Search (MSA-MCS)

To address the SSCF problem, a new hybrid CS scheme called *Memetic Modified Cuckoo Search* is proposed. Now, the exploration-exploitation trade-off is achieved through the combination of two techniques:

1. We adopt the MCS method for global optimization. However, instead of the adaptive method in Eq. (8), we consider a new strategy to modify α dynamically, given by:
$$\alpha^{t+1} = \alpha^t \, Exp\left[-2\pi\left(\frac{t-1}{\Lambda}\right)\right] \tag{9}$$

where Λ denotes the maximum number of iterations. The main difference between both strategies is that the values for α at early iterations are larger for Eq. (9), and the opposite for last iterations (i.e., Eq. (9) boosts a larger exploration at early stages and a larger exploitation at late stages).

2. This global-search technique is then hybridized with a local-search heuristics: the *Adaptive Step Size Random Search* technique [16]. It is based on the idea of changing adaptively the radius of the hypersphere around the most promising solutions for higher accuracy and to escape from local optima. Roughly, the method starts by sampling two points from a hypersphere surrounding the most promising solutions (using Marsaglia's technique [14]). These two points are sampled at different radius, the current one and a larger step in each iteration; the larger is accepted whenever it leads to an improved result. If neither of the two step values lead to improvement for several iterations in a row, smaller step sizes are taken, and the algorithm continues.

Of course, these new features introduce new control parameters in our method, that have also to be properly tuned. This issue will be discussed in Section 4.3.

4.2 Application to the SSCF problem

Given a 2D self-similar binary fractal image $\mathcal{I}^{\square}$, we apply the MSA-MCS method to solve the SSCF problem. We consider an initial population of χ individuals $\{\mathcal{C}_i\}_{i=1,\dots,\chi}$, where each individual $\mathcal{C}_i = \{\mathcal{C}_i^{\kappa}\}_{\kappa}$ is a collection of η real-valued vectors $\mathcal{C}_{\kappa}^{i}$ of the free variables of Eq. (1), as:

$$\mathcal{C}_{\kappa}^{i} = (\theta_{1,1}^{\kappa,i}, \theta_{1,2}^{\kappa,i}, \theta_{2,1}^{\kappa,i}, \theta_{2,2}^{\kappa,i} | \sigma_{1}^{\kappa,i}, \sigma_{2}^{\kappa,i} | \omega_{\kappa}^{i}) \tag{10}$$

These individuals are initialized with uniform random values in $[-1, 1]$ for the variables in $\mathbf{\Theta}_{\kappa}$ and $\mathbf{\Sigma}_{\kappa}$, and in $[0, 1]$ for the ω_{κ}^{i}, such that $\sum_{\kappa=1}^{\eta} \omega_{\kappa}^{i} = 1$. After this initialization step, we compute the contractive factors μ_{κ} and reinitialize all functions ϕ_{κ} with $\mu_{\kappa} \geqslant 1$ to ensure that only contractive functions are included in the initial population. Before applying our method, we also need to define a suitable fitness function. Different metrics can be used for our problem. The most natural choice is the Hausdorff distance, but it is computationally expensive and inefficient for this problem. In this paper the Hamming distance is used instead: we consider the fractal images as binary bitmap images on a grid of pixels for a given resolution defined by a mesh size parameter, m_s. This yields matrices with 0s and 1s, where 1 means that the pixel is activated and 0 otherwise. Then, we count the number of mismatches between the original and the reconstructed matrices to determine the similarity error rate between both images. Dividing this value by the total number of active pixels in the image yields the *normalized similarity error rate* (NSER). This is the fitness function used in this paper.

4.3 Parameter Tuning

The parameter tuning of metaheuristics is slow, difficult, and problem-dependent. Fortunately, the cuckoo search is specially advantageous in this regard, as it only

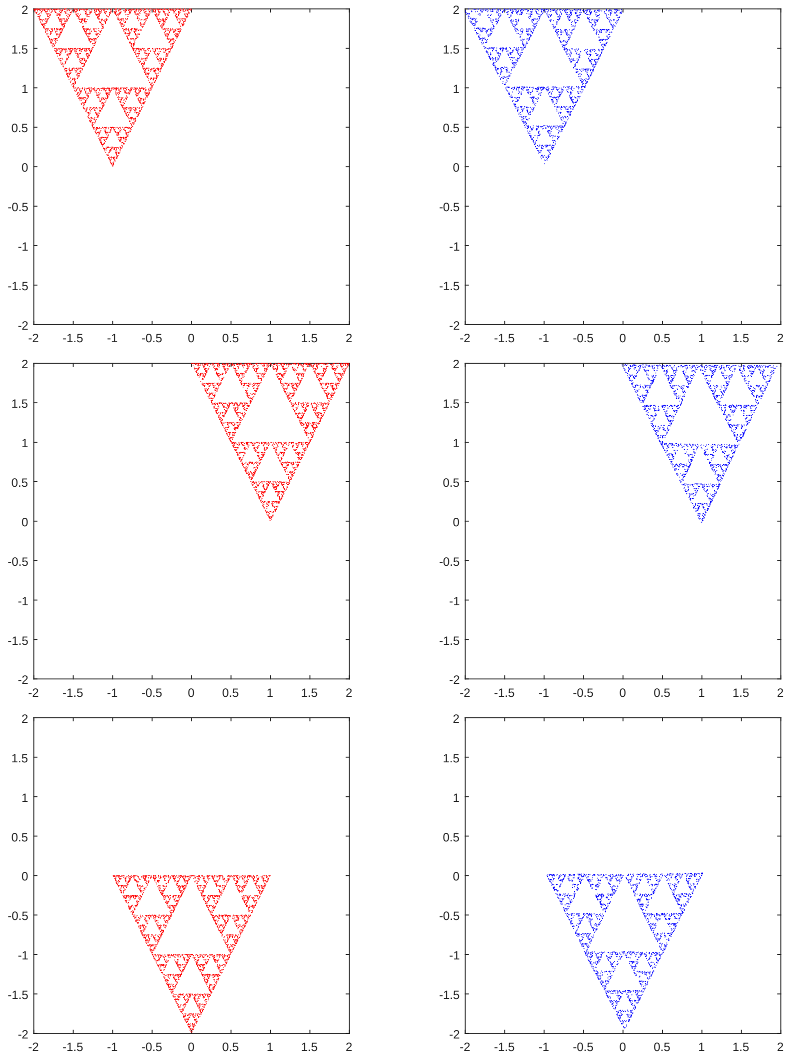

Fig. 1. Graphical results for the Sierpinsky gasket fractal: (left) original images of the three contractive functions; (right) reconstructed images with the MSA-MCS method.

depends on two parameters: the population size, χ, and the probability p_a. We carried out some numerical trials for different values of these parameters and found that $\chi = 40$ and $p_a = 0.25$ are very adequate for our problem. However, the MCS also requires three additional parameters: the initial step size for the Lévy flights, α^0, the number of nests to be abandoned, ρ, and the fraction of nests to make up the top nests, τ. Following some previous works, they have been set to $\alpha^0 = 1$, $\rho = 0.75$ and $\tau = 0.25$, respectively. Moreover, the method is

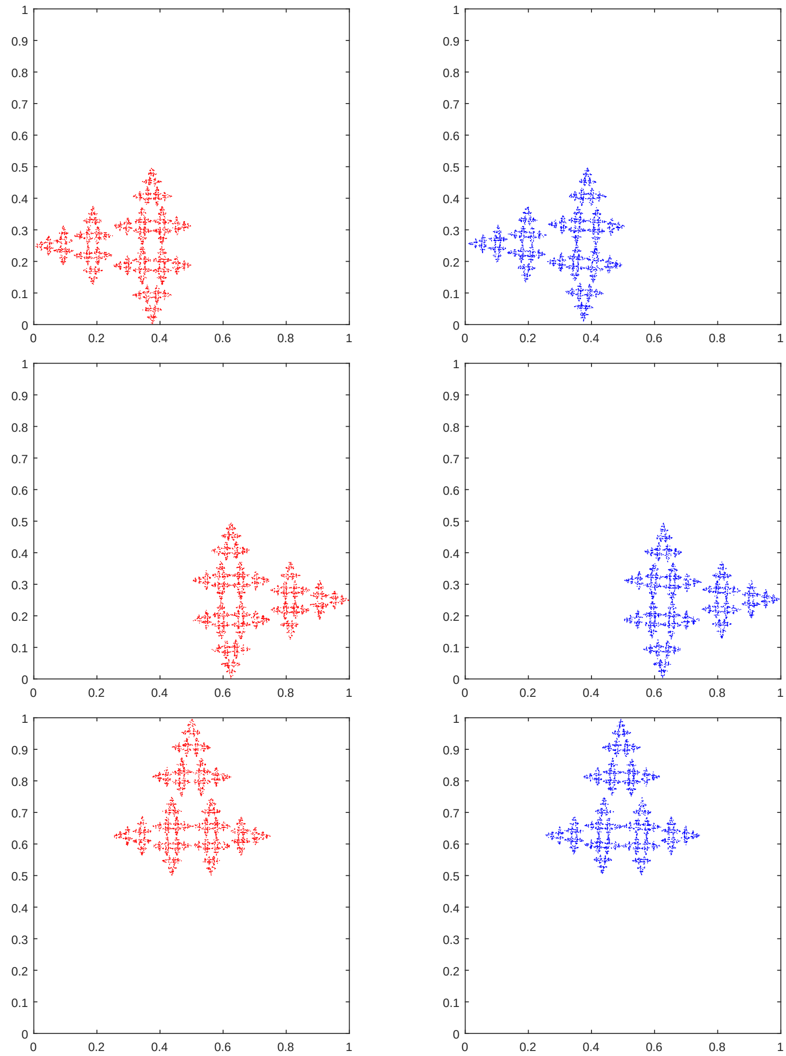

Fig. 2. Graphical results for the Christmas tree fractal: (left) original images of the three contractive functions; (right) reconstructed images with the MSA-MCS method.

executed for Λ iterations. In our simulations, we found that $\Lambda = 2500$ is enough to reach convergence in all cases. In addition to the control parameters for our method, we also need two more parameters related to the problem: the number of contractive functions η and the mesh size, ν. In this work, they are set to $\eta = 3$ and $\nu = 40$, respectively. Unfortunately, we cannot analyze here how all our parameters affect the method performance because of limitations of space.

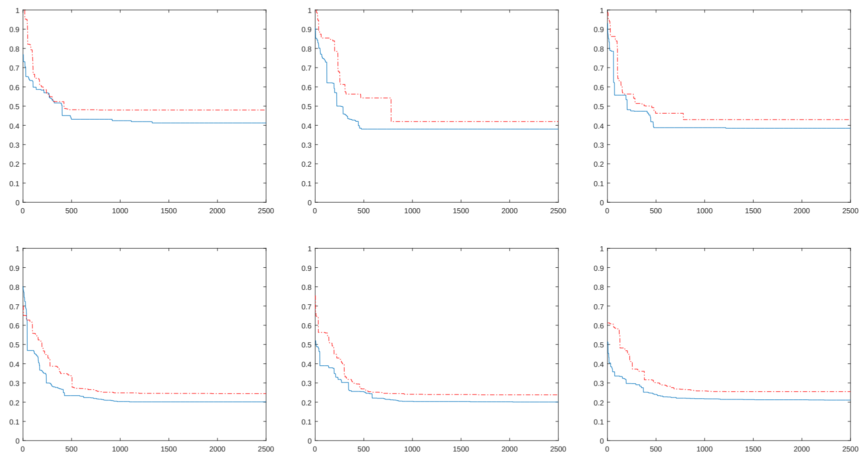

Fig. 3. Convergence diagram of the normalized similarity error rate for the three contractive functions (left to right) of the Sierpinsky gasket (top) and the Christmas tree (bottom) with the original CS algorithm (red dashed line) and our MSA-MCS method (blue solid line).

Table 2. Numerical results of the normalized similarity error rate for the three contractive functions of the examples in Figs. 1 and 2 with the original CS algorithm and our MSA-MCS method (see also Fig. 3 for their graphical representation).

	Sierpinsky gasket			*Christmas tree*		
	NSER(ϕ_1)	NSER(ϕ_2)	NSER(ϕ_3)	NSER(ϕ_1)	NSER(ϕ_2)	NSER(ϕ_3)
Best (CS):	0.4798	0.4178	0.4296	0.2445	0.2382	0.2547
Mean (CS):	0.4992	0.4333	0.4501	0.2603	0.2511	0.2769
Best (MSA-MCS):	0.4124	0.3805	0.3849	0.2014	0.2008	0.2113
Mean (MSA-MCS):	0.4403	0.4157	0.4195	0.2206	0.2257	0.2331

5 Experimental Results

All computations in this paper have been performed on a 2.6 GHz. Intel Core i7 processor with 16 GB. of RAM. The source code has been implemented by the authors in the native programming language of the popular scientific program *Matlab version 2015a* and using the numerical libraries for fractals in [3, 5–7]. Our method has been applied to several examples of fractals with $\eta = 3$. Only two (already analyzed in [13]) are included here because of limitations of space: the Sierpinsky gasket and the Christmas tree, depicted in Figs. 1 and 2, respectively. The figures show the fractal images of the original (in red) and the reconstructed (in blue) contractive functions on the left and the right columns, respectively. The images correspond to the best value of the NSER fitness function selected from a set of 50 independent executions. As shown in the images, our MSA-MCS approach captures the structure and general shape of the contractive functions with high visual quality. This is a remarkable result because our initial population is totally random, meaning that their corresponding images are all very far from the given fractal image. Figure 3 shows the convergence diagram for the three contractive functions (from left to right) of the Sierpinsky gasket (top row) and the Christmas tree (bottom row) using the original CS method (as reported in [13]) and the new MSA-MCS method, displayed as red dashed lines and blue solid lines respectively. This figure shows that the new method MSA-MCS outperforms the previous CS method for all contractive functions of both examples.

The good visual appearance of the method in Figs. 1-2 and its graphical comparison with the CS method in Fig. 3 are all confirmed by our numerical results reported in Table 2. The table shows the best and the mean values of the normalized similarity error rate, NSER(ϕ_κ), for 50 independent runs. These results indicate that the new MSA-MCS method performs quite well. It also improves the previous results in [13] based on the original CS algorithm by a significant margin in all cases. For instance, we can see that the even the mean value of NSER for MSA-MCS is better that the best value of NSER with the original CS method. In other words, it is not a case of just an incremental improvement, but a significant one statistically.

6 Conclusions and Future Work

In this paper we address the problem to compute the optimal set of individual contractive functions for the reconstruction of self-similar binary fractal images. To this aim, we propose a memetic approach comprised of the modified CS method for global optimization with a new strategy for the Lévy flight step size (MMCS) and the ASSRS heuristics for local search. This approach is applied to some illustrative examples of fractal images with satisfactory results. This new method shows a significant improvement with respect to a previous approach based on the original CS for all functions in our benchmark.

In spite of these good results, there is still room for further improvement in the SSCF problem. We also wish to address the second step of the general IFS inverse problem for self-similar fractal images and its extension to the case of non self-similar fractals. We also plan to apply a very promising recent hybrid self-adaptive cuckoo search [15] to our problem as part of our future work.

Acknowledgements
This research is supported by the project PDE-GIR of the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 778035, the Spanish Ministry of Economy and Competitiveness under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds FEDER (AEI/FEDER, UE), the project #JU12, of SODERCAN and European Funds FEDER (SODERCAN/FEDER UE) and the project EMAITEK of the Basque Government.

References

1. Barnsley, M.F.: Fractals Everywhere (2nd Ed.). Academic Press, San Diego (1993).
2. Falconer, K.: Fractal Geometry: Mathematical Foundations and Applications (2nd Ed.). John Wiley & Sons, Chichester, England (2003).
3. Gálvez, A.: IFS Matlab generator: a computer tool for displaying IFS fractals. In: Proc. ICCSA'2009, pp. 132–142. IEEE CS Press, Los Alamitos, CA (2009).
4. Gálvez A., Iglesias A.: Cuckoo search with Lévy flights for weighted Bayesian energy functional optimization in global-support curve data fitting. *The Scientific World Journal*, 2014, Article ID 138760, 11 pages (2014).
5. Gálvez, A., Iglesias, A., Takato, S.: Matlab-based KETpic add-on for generating and rendering IFS fractals. CCIS, 56, 334–341 (2009).
6. Gálvez, A., Iglesias, A., Takato, S.: KETpic Matlab binding for efficient handling of fractal images. Int. J. Future Generation Comm. & Networking, 3(2), 1–14 (2010).
7. Gálvez, A., Kitahara, K., Kaneko, M.: IFSGen4LaTeX: Interactive graphical user interface for generation and visualization of iterated function systems in LaTeX. LNCS 8592, 554–561 (2014).
8. Gutiérrez, J.M., Iglesias, A.: A Mathematica package for the analysis and control of chaos in nonlinear systems. Computers in Physics, 12(6), 608–619 (1998).
9. Gutiérrez, J.M., Iglesias, A., Rodríguez, M.A.: A multifractal analysis of IFSP invariant measures with application to fractal image generation. Fractals 4(1), 17–27 (1996).
10. Gutiérrez, J.M., Iglesias, A., Rodríguez, M.A., Burgos, J.D., Moreno, P.A.: Analyzing the multifractal structure of DNA nucleotide sequences. Chaos and Noise in Biology and Medicine 7, pp. 315–319. World Scientific, Singapore (1998).
11. Hutchinson, J.E.: Fractals and self similarity. Indiana Univ. Math. Journal 30(5), 713–747 (1981).
12. Iglesias, A., Gálvez, A.: Cuckoo search with Lévy flights for reconstruction of outline curves of computer fonts with rational Bézier curves. Proc. of Congress on Evolutionary Computation-CEC'2016. IEEE CS Press, Los Alamitos, CA (2016).
13. Quirce, J., Gálvez, A., Iglesias, A.: Computing self-similar contractive functions for the IFS inverse problem through the cuckoo search algorithm. Proc. Int. Conf. on Harmony Search Algorithm, ICHSA'2017. Springer, Advances in Intelligent Systems and Computing, 514, 333–342 (2017).
14. Marsaglia, G.: Choosing a point from the surface of a sphere. Annals of Mathematical Statistics, 43(2), 645–646 (1972).
15. Mlakar, U., Fister Jr., I., Fister, I.: Hybrid self-adaptive cuckoo search optimization. Swarm and Evolutionary Computation, 29, 47–72 (2016).
16. Schumer, M.A., Steiglitz, K.: Adaptive step size random search. IEEE Transactions on Automatic Control, 13(3), 270–276 (1968).
17. Walton, S., Hassan, O., Morgan, K., Brown, M.R.: Modified cuckoo search: a new gradient free optimisation algorithm. Chaos, Solitons & Fractals, 44, 710–718 (2011).
18. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms (2nd. Edition). Luniver Press, Frome, UK (2010).
19. Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: Proc. World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214. IEEE Press, New York (2009).
20. Yang, X.S., Deb, S.: Engineering optimization by cuckoo search. Int. J. Mathematical Modelling and Numerical Optimization 1(4), 330–343 (2010).