Time optimizing of fractal image compression using the Scatter Search algorithm

Shaimaa S. Al-Bundi
Department of Mathematics
College of Education for Pure Sciences
Ibn Al-Haitham
University of Baghdad
shaimaa.s.a@ihcoedu.uobaghdad.edu.iq

Nadia M.G. Al-Saidi*
Department of Applied Sciences
University of Technology
Iraq
nadia.m.ghanim@uotechnology.edu.iq

Abstract. Image compression is one of the essential requirements for the efficient use of storage space and bandwidth. A new technique based on fractal theory is proposed for encoding the image; it is known as fractal image compression. In the procedure of encoding, the mechanism of search is considered as one of the main problems of this technique. In this work, an attempt to speed up the encoding process with minimal loss of the compressed image quality is adopted based on the Scatter Search algorithm. It is a sibling of Tabu search based on similar origins. The experimental results show a significant reduction in the computation time, where the mean square error measures between blocks are decreased after comparing them to full search methods. Consequently, the decoding process evinced that the reconstructed images were of high quality.
Keywords: fractal image coding (FIC), iterated function system (IFS), optimization algorithm, metaheuristics, scatter search method.

1. Introduction

Fractal image compression (FIC) is demonstrated to deliver an excellent compression ratio, and high quality reconstructed images. These two main significant concepts attract researcher attention since this technique was discovered by Barnsley in 1988 [1]. His method is based on the theory of iterated function system (IFS) [2, 3]. He obtained around 1000:1 compression ratio, but it required manual interference. This problem motivates his student Jacquin to improve this technique to be functioned automatically through partitioning the image into blocks [4, 5]. His approach focused on how efficiently can find the appropriate domain block for each range block, which in turn, given the approximate transformation after partitioning the image into a non-overlapping rang block and an overlapping domain block,. This process is known as an encoding

*. Corresponding author

method. The attractor (fractal) is constructed by the decoding process based on the fixed point theorem, starting with any initial image. However, Jacquin's approach is considered as the main core of this research direction. Despite the encouraging properties of this high compressed technique, it has a significant constraint, and this is due to its intensive computation required in searching for the matched areas in the image. This limitation motivates the scholarly to trend their efforts toward overcoming such problem and optimizing the time performance of the encoding process.

Focusing on two main categories in the FIC technique, which are; finding the IFS, and an efficient algorithm to construct the approximate corresponding IFS for the given image, makes it completely different from conventional image compression techniques. After the backbone of FIC via the IFS, which was introduced by Barnsley and Sloan [6, 7], several attempts are highlighted toward improving the search for matching blocks or the type of segmentation utilizing different approaches [8, 9, 10, 11, 12, 13, 14, 15, 16]. The optimization algorithm is proved to be efficient for such a purpose and considered as a significant objective for many authors, because it helps in increasing the abilities to match computation. Based on evolutionary computation methods, many contributions are found in recent literature, some of them are based on the natural inventor, and the others are bio-inspired [17, 18, 19, 20, 21, 22, 23, 24, 25].

Some of the literatures that based on evolutionary methods were proposed to increase the compression ratio and preserve the quality of the decoded fractal attractor to generate the compressed image. The optimization is performed by the mechanism the number of selection that achieves a high reduction in the time, which may approach 150, as well as, a sound decrease in mean square error that may approach 66%. Examples of such literatures in this context are; genetic algorithm [19, 20, 21, 22], crowding method [23], ant colony [24], particle swarm optimization [25, 26], harmony search method [27]. Also, the one that used the heuristic algorithm based on the Wolf Pack algorithm was introduced in 2017 [28]. This algorithm works by exploring the whole space looking for similar small blocks. Another improvement is achieved in [29, 30] by adopting of Cuckoo inspired algorithm which arrange the range blocks as an ordered vector by the similarity measure or coordinate distance.

In contrast to these methods and association with the evolutionary computation methods, we adopted in this work the scatter search (SS) method [31, 32] as one of the metaheuristics and a global optimization algorithm utilized to reduce the searching strategy. Scatter Search is a sibling of Tabu search [33, 34]. This searching strategy is introduced to combine decision rules in the context of scheduling techniques, and to obtain improved decision-making rules for scheduling problems [5]. The purpose of this approach is to assume that the information about the relative desire for alternative options can be achieved in different forms through different rules. This information can be used more effectively when it is integrated with the combination mechanism and processed through the standard strategy by selecting different rules at the same time. By the SS, the bset is continually updating; its update depends on the quality and diversity of the solutions. In each updated set, the weak solutions that do not give the desired results are eliminated after calculating the least square error.

Besides all of these advantages of the scatter search method, it provides precise and fast results to exclude the redundant blocks with a shortening time. Therefore, it was used to speed up the fractal image compression technique.

Several multimedia and communication applications are based on image compression, which plays an essential role in evolving such services. Consequently, a small size image can be transmitted quickly. Fractal techniques for image compression are proved to be the best over other techniques due to their ability to remove redundancy and maintain image resolution. However, the problem with this technique is time-consuming. The main contribution of the present work is to utilize the SS algorithm; it works toward improving the solution and speeding up the matching process to exclude the redundant areas.

The rest of the paper is organized as follows: Section 2 presents the original algorithms of the scatter search method. Section 3 describes the proposed fractal image compression based on the scatter search method. Section 4 discusses and analyses the implemented, and the results in terms of some statistical measurement. Conclusions are summarized in Section 5.

2. Scatter search method

As a nature-inspired method, the scatter search method, in combination with the sibling of Tabu search, was introduced in 1977 by Glover [31]. He used it to maintain a good quality nominates solution and diversity set based on the principle that the vital information for the global optima is saved in a reference set of solutions. After that, this information is exploited by the recombining of some other samples. This procedure is performed through the iteration process, and it is highly used for reducing the cost. Glover was proposed many literature for different purposes [31, 32, 33, 34, 35]. He proved that the dispersion patterns generated by SS are useful in various application areas and applied to many optimization problems. Its mechanism emphasizes on creating of a new solution based on the basic set (bset) that represents the combinations of high-quality solutions with diverse solutions, where the bset represents a weighted of the combination to expatiate the optimization problem directly. By the SS method, it is evident that the randomness degree is lower than those appears in the meta-heuristic method that emphasized on the population, as well as, using the local search method to speed up the convergence to the optimum range is considered as another differences between the SS and other conventional methodologies that based on the population.

The main three parts of this method are:

1. Generate a collection of diverse trial solutions.

2. The local search, contain the bset.

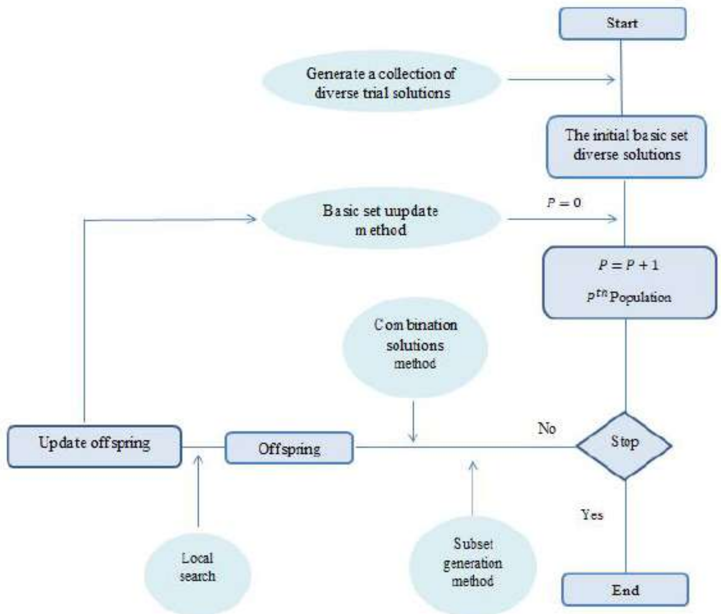


Figure 1: Scatter Search Steps

3. Update the bset to create and preserve new bset that consists the best solution.

   These parts represent the initialization stage, while the other that used for search purposes are:

4. The method of configuring a subset that operated on bset is to create a subset of solutions to be used as a basis for producing collective solutions.

5. The method of combined solutions is used to convert a given subset of solutions produced by the subset creation way into one or more collected solution vectors, Glover (1999) [32]. Figure 1 illustrates the five scatter search steps.

The basic set arrangement and repetition: By this process, the elements of the bsetare sorted based on their quality. When some elements are neighbors and close to each other, it should be replaced by a random solution, as can be seen in steps (8-12, **Algorithm 1**). This comparison process is applied for each pair in the bset. Besides, step 9 illustrates how two solutions are close to each other if the maximum distance between them is more significant than a given threshold. However, this step leads to increase the diversity of the bset and prevent the stagnation of the search process.

Basic set= $\{\alpha^1, \alpha^2, \ldots, \alpha^m\}$, bset $\subset P, f(\alpha^i) \leq f(\alpha^j), i, j \in \{1, 2, \ldots, m\}$, $i < j$, where P represents the population, m is the size of the population, and f is the nonlinear dynamic problem.

If $\max|\frac{\alpha^i-\alpha^j}{\alpha^j}| \leq \epsilon, i < j$ then change α^j in the *bset* by a random solution to be evaluated, such that:$\Rightarrow$ $eval = eval + 1$ *Generating of new solutions*: This process is performed by the steps (13-23, Algorithm 1), the new solution is generated in a particular rectangles according to its position, and the distance between the elements. It is accomplished by combining all the dimension of the boundaries with a random weight and bounded by the combined solution distance, such that;

$$\beta = \beta \cup \beta^{i,j} \text{ and evaluate } \beta^{i,j} \ \Rightarrow eval = eval + m - 1$$

Updating of the bset: The solution created by the combination of the solution can replace the bset members, as shown in step (26, **Algorithm 2**), if they exceed the predefined threshold. Otherwise, the solution cannot replace the members (parent); it should replace the original elements in the primary group. This technique avoids combinations of identical solutions in the early stages of algorithm that can lead to premature inactivity in the search process.

Replace classified bset: The best children $\beta^{i,*}$replaces the classified *bset* members then reset $n(i)$.

After that, two additional techniques are used to make the search more efficient (**Algorithm 2**). The first technique (go-beyond) is to use the new solution if it exceeds the original and a new solution is created in an excessive rectangle, this process is repeated as long as there are improved solutions as shown in steps (1-26, **Algorithm 2**).

Let $\beta^{i,*} = best\ children \in \beta^{i,j} \forall j \in \{1, 2, \ldots, m\}, i \neq j$.

If $\beta^{i,*} < f(\alpha^i)$ then create a new solution α_{new} in the hyper-rectangle, such that:

$\alpha_{new} = \left(\beta^{i,*} - \frac{\alpha^i - \beta^{i,*}}{\Omega}, \beta^{i,*}\right), \Omega = 1$, where Ω depends on the number of improvements in **Algorithm 2**, and $\beta^{i,*} = \alpha_{new}$.

If $\left(\beta^{i,*}\right) < fbest$ then $\alpha\ best = \beta^{i,*}$, and $fbest = f\left(\beta^{i,*}\right)$.

The stagnation exam: The second technique includes the stagnation exam, it is considered as a local solution, and it should be replaced by a new random solution in the *bset* as illustrated in steps (27-32, **Algorithm 2**). Let $n(i)$ be the counter that replace the classified bset members by best children $\beta^{i,*}$ $n(j) = n(i) + 1$, where j is the index of the unclassified *basicset* members If $n(i) > nchange$ then change $\alpha^i \in bset$ by the random solution and set $n(i) = 0$.

The evaluation process: This process is performed by testing the function of local search $LS(w_1)$, if its value is less than or equal to an evaluator tolerance, then a local search over $\alpha bset$ is applied. Otherwise, the function $LS(w_2)$ is checked with the same tolerance to determine the previous local search. However, such a local search can only be applied to the best existing solutions that have been updated in the current iteration, as illustrated in the following steps.

Let *eval* denotes the evaluator of α^i. If $eval \leq LS(w_1)$ then the local search is applied over α *best* sol.

If eval $\geq LS\left(w_2\right)$ then create $\boldsymbol{\beta}=\left\{\beta^1, \beta^2, \ldots, \beta^m\right\}$ such that $f\left(\beta^i\right) \leq f\left(\beta^j\right)$ if $i<j$ $d_{\min}(i)=\left|\beta^i-LS\right|^2, i \in\{1,2, \ldots, m\} \subset \beta$ and LS refers to the local solution, where all the local optima are found. Sort $\boldsymbol{\beta}$ according to their diversity, creating $\boldsymbol{\beta}_d=\left\{\beta_d^1, \beta_d^2, \ldots, \beta_d^m\right\}$, such that $d_{\min}(i) \geq d_{\min}(j)$.

The distance between the new and the previous solutions is used to measure the diversity and the balance between their diversity and quality. The newly obtained solution is added to the set of local solutions.

In the following equation the parameter balance is used as a measure of quality or diversity by giving them a weight, score $\left(\beta^k\right)$ represents the counter for this balance such that: score $\left(\beta^k\right)=(1-\text{balance}) * i+(\text{balance} * j)$, where i is the index of $\beta^k \in \boldsymbol{\beta}$ and j is the index of $\beta^k \in \boldsymbol{\beta}_d$, and $i<j$ Now, apply local search over β^l, such that score $\left(\beta^1\right)=\min(\text{score}(\boldsymbol{\beta}))$. If $X^* \notin LS \Rightarrow LS$ $=LS \cup\left\{X^*\right\}$.

Algorithm 1 Classical Scatter Search

Set parameters: eval, Local solutions (LS), diverse, bset, dimension solutions(m), $LS(w_1), LS(w_2)$, balance, score best solution
Initialize n, $eval$;
$LS=\phi$;
Generate a collection of diverse trial solutions P;
$eval = eval + diverse$;
Create the initial bset of dimension solutions m with the best solutions and random elements.
Repeat
Arrange bset by quality, such that bset $=\{\alpha^1, \alpha^2, \cdots, \alpha^m\}$, and $f(\alpha^i) \leq f(\alpha^j)$, where $i, j \in \{1,2,\cdots,m\}, i<j$.
if $\max\left|\frac{a^i-a^j}{a^j}\right| \leq \epsilon, i<j$ **then**
  Put α^j in the bset by a random solution and evaluate it;
  $eval = eval + 1$;
end if
$\beta=\phi$;
for i=1 to m **do**
  for j=1 to m **do**
    if $i \neq j$ **then**
      Combine α^i with α^j to generate a new solution $\beta^{(i,j)}$;
      $\beta=\beta \cup \beta^{(i,j)}$;
      Evaluate $\beta^{(i,j)}$;
    end if
  end for
  $eval = eval + m - 1$;
end for

Applied algorithm of additional techniques (**Algorithm2**)
Until a stopping criterion is met to best found solution or set of solutions.

In Algorithm 2, the two methods are used to make the search for an improvement solution more efficient. It is based on two techniques. First, the enhanced solution is used instead of the original solution, which can be shown in steps (1-26). In the second, the local solution is replaced by the random solution from the solutions set; this can be seen in steps (27-32).

Algorithm 2 Additional Techniques

1: **for** i=1 to m **do**
2: $\beta^{(i,*)}$ = best children in $\beta^(i, j)\forall j \in \{1, 2, \ldots, m\}, i \neq j$;
3: **if** $(\beta^{(i,*)}) < f(\alpha^i)$ **then**
4: $improvement = 1$;
5: $\Omega = 1$; do
6: Create a new solution, α_{new} in the hyper-rectangle.
7: $\alpha_{new} = \left(\beta^{i,*} - \frac{a^i - \beta^{L*}}{\Omega}, \beta^{i,*}\right)$;
8: $\beta^{(i,*)} = \alpha_{new}$
9: $eval = eval + 1$;
10: $\beta = \beta \cup \beta^{(i,*)}$;
11: $improvement = improvement + 1$;
12: **if** $improvement = 2$ **then** then
13: $\Omega = \frac{\Omega}{2}$;
14: $improvement = 0$;
15: **if** $f(\beta^{(i,*)}) < fbest$ **then**
16: $\alpha\ best = \beta^{(i,*)}$;
17: $fbest = f(\beta^{(i,*)})$;
18: **end if**
19: **end if**
20: **end if**
21: **end for**
22: **if** the local search is activated **then**
23: Applied algorithm of local search (3)
24: **end if**
25: Replace classified basic set members by best children $\beta^{(i,*)}$ and reset $n(i)$;
26: $n(j) = n(i) + 1$; j is the index of a not classified basic set members.
27: **for** $i = 1tom$ **do**
28: **if** **then**$n(i) > nchange$

Replace $\alpha^i \in basic$ set by a random solution and set $n(i) = 0$;
end if
end for

Algorithm 3 The Local Search

if $LS = \phi$ **then**
if $eval \geq LS(w_1)$ **then**
Apply local search over α best;
end if
end if
elsif
if $Localsolutions(LS) = \phi$ **then**
if $eval \geq LS(w_2)$ **then**
Sort β by quality, create $\beta = \{\beta^1, \beta^2, \cdots, \beta^m\}$, such that $f(\beta^i) \leq f(\beta^j)$ if $i < j$.
Calculating the minimum distance between each element $i \in 1, 2, \ldots, m \subset \beta$, and all the local optima that are found.
$d_{min}(i) = |\beta^i - LS|^2$;
Sort β by diversity, creating $\beta = \{\beta_d^1, \beta_d^2, \ldots, \beta^m\}$, such that $d_{min}(i) \geq d_{min}(j)$;
if $i < j$ **then**
for each solution $\beta^k \in \beta$ **do** score(β^k) $= (1 - balance) * i + balance * j$, such that i is the index of $\beta^k \in \beta$, and j is the index of $\beta^k \in \beta_d$;
end for
Apply local search over β^1, such that score $(\beta^1) = \min(score(beta))$;
end if
end if
end if
Create the Local Solution X^*;
if $X^* \notin LS$ **then**
$LS = LS \cup \{X^*\}$;
end if
if $f(X^*) < fbest$ **then**
$\alpha\ best = X^*$;
$fbest = f(X^*)$;
end if

3. The Scatter Search method to improve Fractal Image Compression technique

In this section, FIC technique is implemented using SS method improvement was accomplished using SS algorithm is one of the population-based Metaheuristics.

The classical Jacquin approach is began by partitioning the given image into non-overlapping blocks called range blocks (RBs) $R = \{R^1, R^2, \ldots, R^m\}$ of size $r \times r$ and overlapping domain blocks blocks (DBs), $D = \{D^1, D^2, \ldots, D^n\}$of size $2r \times 2r$, where $m = (\frac{M}{r})^2$, and $n = (M - 2r + 1)^2$. For each R^i, search for its similar $D^k, k = 1, \ldots, n$ with an appropriate contractive transformation w^{ik} that persuaded the equation $d\left(R^i, w^{ik}\left(D^k\right)\right) = \min d\left(R^i, w^{ij}\left(D^j\right)\right)$. Here, w^{ik} represents the contractive affine transformation between D^j to R^i. This can be represented by $d\left(R^i, w^{ij}\left(D^j\right)\right)$, where d is the MSE between R^i and $w^{ij}\left(D^j\right)$, which in turn consist of two mappings T^j and θ^{ij} such that, $w^{ij} = \theta^{ij} * T^j$. Meanwhile, T^j represents the scaling transformation that used to contract D^i size to equal R^i size by averaging of the four pixels in the block of D^j. However, $T^j\left(D^j\right)$ is equal to R^i size as illustrated in Figure 2.

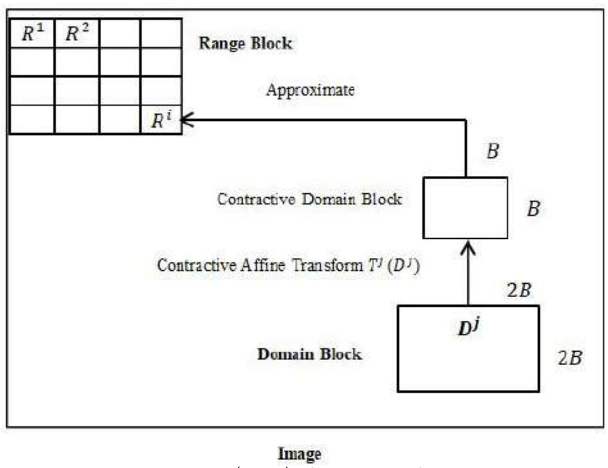


Figure 2: Domain Block (DB) Search of a Range Block (RB)

The mapping ϕ^{ij} is subjected to the eight isometries transformations given in Table 1.

Subsequently, W^{ij}, is used to transform the pixel values of the transformed block, such that:

$$W^{ij}(z) = s^i z + o^i. \tag{1}$$

The variables z, s^i,and o^i in Eq.(1) represent the pixel value, scaling, and the offset o^i respectively, where s^i,and o^i are obtained by Eq.s (2 and 3) respectively. These two values are used to calculate the MSE.

$$s = \frac{\left[n \sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} b_i\right]}{\left[n \sum_{i=1}^{n} a_i^2 - \left(\sum_{i=1}^{n} a_i\right)^2\right]}, \tag{2}$$

$$o = \frac{1}{n}\left[\sum_{i=1}^{n} b_i - s \sum_{i=1}^{n} a_i\right]. \tag{3}$$

Table 1: The eight isometries

Transform T^j	Angle	Isometry
T^0	Identity	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
T^1	Rotation (+90°)	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$
T^2	Rotation (+180°)	$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$
T^3	Rotation (+270°)	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$
T^4	Reflection abt x axis	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$
T^5	Reflection & Rotation (+90°)	$\begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$
T^5	Reflection & Rotation (+180°)	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
T^5	Reflection & Rotation (+270°)	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

The MSE difference is used to measure the distances between the solutions, as computed in Eq.(4):

$$MSE = \frac{1}{n}\left[\sum_{i=1}^{n} b_i^2 - s\left(s\sum_{i=1}^{n} a_i^2 - 2\sum_{i=1}^{n} a_i b_i + 2o\sum_{i=1}^{n} a_i\right)\right.$$
$$\left. +o\left(n^* o - 2\sum_{i=1}^{n} b_i\right)\right]. \tag{4}$$

This equation is used to calculate the scaling as a geometric part and the image pixels as a massive part. By using scatter search, a random search is performed in the selected community $P = \{P_1, P_2, \ldots, P_m\}$, the local search for (initial P) is then conducted. Then calculated the difference between the best children (best solution) $\beta^{i,*}$ and w^{ij}

$$d = \sum \left|\beta^{i,*} - w^{ij}\right|, \tag{5}$$

where $w^{ij} = \theta^{ij} * T^j$ and $T^j \in P(P$ is collection of diverse trial solutions). Updating the bset by choosing the best solution after tests, where bset update method equal high quality union the diversity then the best solution $\beta^{i,*}$ is chosen. The worst solution from the bset is removed and again working to get improved solutions by applying steps 7 and 8 in Algorithm 4 Finally, stop the processing when getting the best solution or the loop of find the solution is end. Algorithm 4 describes fractal image compression based on scatter search algorithm:

Algorithm 4 The Proposed Algorithm

1: Initialize the scatter search;
2: Split the given image M into two types of blocks, one of them is called range blocks $RB = \{R^1, R^2, \ldots, R^m\}$ of size $\times r$, which are non-overlapping, where $m = (\frac{M}{r})^2$, and the other is called domain blocks $DB = \{D^1, D^2, ..., D^n\}$ of size $2r \times 2r$ blocks of overlapped area, on which $n = (M - 2r + 1)^2$.
3: For each range block R^1, choose the domain block $D^k, k = 1, \ldots, n$ and a suitable contractive affine transformation w^{ik} that satisfied $d(R^i, w^{ik}(D^k)) = mind(R^i, w^{ij}(D^j))$, where $w^i k$ is a contractive affine transformation, $w^{ik} : D^i \to R^i$. Here d represents the MSE between R^i, and w^{ik}, where $w^i j = \theta^{ij} * T^j$. The function $T^j(D^j)$ is carrying out using eight isometries given in Figure 2.
4: The pixel values of the output values after implementing of the first step is transformed by W^{ij}, such that, $w^{ij}(z) = s^i z + o^i$, where s^i and o^i are given in equations (2) and (3) respectively.
5: This comparison resulting $T^j = [x^i, y^i, w^i, s^i, o^i]$;
6: The scatter search affects the above steps as follows:
7: Generate a collection of diverse trial solutions $P = (P_1, P_2, \ldots, P_m)$. However, by the scatter search, a population of 10 samples is generated in each time.
8: Create the initial bset of dimension solutions equals m with the best solutions and random elements.
9: **Repeat**
10: Arrange the bset according to its quality, such that $bset = \{T^1, T^2, \ldots, T^m\}$. Hence, $w(T^i) \leq w(T^j)$, where $i, j \in \{1, 2, \cdots, m\}, i < j$
11: Determine the LS for (initial P)
12: $d = \sum |\beta^{(i,*)} - w^{ij}|, i, j = 1, 2, \cdots, m$, and $i < j$;
13: The new $bset = highquality \cup diversity$;
14: Apply local search iteration
15: **Repeat**
16: Choose the best solution $\beta^{(i,*)}$, where $\beta^{(i,*)} \in \beta$;
17: Calculate the minimum distance between each element w^{ij} with the local solution,
18: $d_{min}(i) = |w^{ij} - LS|^2$;
19: **if** best changed **then**
20: $X^* = bestsolution$
21: **Else**
22: **end if**
23: Remove the worst solutions $\frac{X^*}{2}$ from the bset;

```
Combine T^i with T^j to generate a new solution β^(i,j)
Replace classified bset members by best children β^(i,*)
Make new population size improved solutions applying step 7 and so on
Until a stopping criterion is met to the best solution or set of solutions
```

4. Implementation and analysis

Image compression plays an essential role in the evolution of several multimedia services and communication applications. Images need high storage area. Consequently, decreasing the volume of the transmitted image data leads to speed up the transportation of such data. The techniques of image compression aim to take out the redundant data with preserving of acceptable resolution of the retrieved image. As an emerging technique, FIC which is based on IFS is demonstrated as the best for image compression technology due to its high ability to remove redundancy and to maintain image resolution. The main task of the SS algorithm is to build a community for generating a complete solution based on Sipling of Tabo search and accordingly, it work toward improving the solution. We calculate the fitness for a set of images and then enter each image by 100 generating and divide into 10 generating produces an image and repeat this process for the other 10 generation and so on. The proposed model was applied on four grayscale images with a size 512 x 512, and a range block of size (2×2),(4×4) and(8×8). The SS algorithm was implemented in MATLAB on a PC with the specification: Intel 2.5 GHz Core i7, with 8 MB memory. Some statistical test are used to evaluate the performance of the proposed method, such as, Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Compression Ratio (CR), these are the crucial parameters. These parameters are evaluated for some images, they are defined as follows: Peak-Signal to Noise Ratio (PSNR): It is the metric that used to measure the quality between the original and the cipher image. It is the ratio between the possible maximum signal value and the value of the corrupted noise. The PSNR is usually expressed by the logarithmic function due to the wide dynamic range of numerous signals. It is defined by Eq. (6) as in the following [6]:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MES}\right)(dB). \tag{6}$$

The equation for calculating mean square error (MSE), which is considered as a risk function and used to measure the square error, is given as follows:

$$MSE = \frac{1}{n*m}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(x_{i,j}-y_{i,j}\right)^2. \tag{7}$$

The PSNR for color images is computed at every color plane of the RGB image taking the average of the MSE's of the three colors. Therefore, the PSNR of the

RGB image is; where:

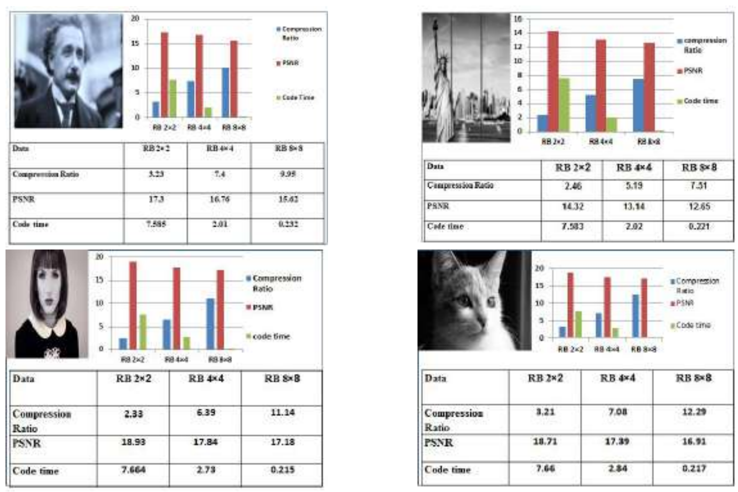
$$PSNR = 10 \log_{10} \left(\frac{255^2}{MES_{RGB}} \right) (dB), \tag{8}$$

$$MSE_{RGB} = \frac{1}{3} \left(MSE_{Red} + MSE_{green} + MSE_{blue} \right). \tag{9}$$

The CR is defined as the ratio of the original image data size over the compressed image data size:

$$CR = \frac{Size\ of\ original\ image}{Size\ of\ compressed\ image}. \tag{10}$$

When the range block size is increased, a low compression time is obtained with a high PSNR, which in turn causes a loss in the quality of the compressed image. Whereas, by decreasing range block size, a high PSNR value is achieved, which refers to high compression ratio with good quality but vise verse time consuming. Therefore, to balance between the PSNR and reasonable computation time besides good compressed image quality, the range blocks is chosen to be of sizes (2×2),(4×4) or (8×8). The experiments show that the decoded image quality of a range blocks of sizes (2×2) and (4×4) are better than that of a size (8×8), and this can be seen obviously in Figure 3.



Data	RB 2×2	RB 4×4	RB 8×8
Compression Ratio	3.23	7.4	9.95
PSNR	17.3	16.76	15.62
Code time	7.585	2.01	0.232

Data	RB 2×2	RB 4×4	RB 8×8
Compression Ratio	2.46	5.19	7.51
PSNR	14.32	13.14	12.65
Code time	7.583	2.02	0.221

Data	RB 2×2	RB 4×4	RB 8×8
Compression Ratio	2.33	6.39	11.14
PSNR	18.93	17.84	17.18
Code time	7.664	2.73	0.215

Data	RB 2×2	RB 4×4	RB 8×8
Compression Ratio	3.21	7.08	12.29
PSNR	18.71	17.39	16.91
Code time	7.66	2.84	0.217

Figure 3: Compression ratio, PSNR and coding time based on the proposed method for different range size

The experimental results show the compression ability that in some images reached 98%. The time consuming of this technique as a weak point motivates us to optimize the computation time by proposing the scatter search method, where the experiments show its efficiency over the classical Jacquin method and also over the IFS based genetic algorithm as shown in Table 2 and Figure 4.

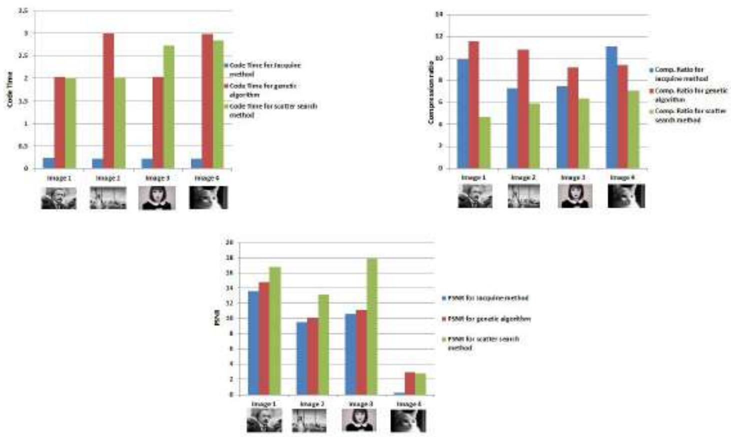


Figure 4: Comparison between Jacquin method, genetic algorithm and the proposed scatter search for range block of size 4.

Table 2: Comparison between Jacquin approach, genetic and the proposed scatter search for range block of size 4.

	Images				
Fractal image compression based on Jacquin method	Compression Ratio	9.97	7.33	7.53	11.16
	PSNR	13.62	9.52	10.65	15.18
	Code Time	0.247	0.226	0.227	0.221
Fractal image compression based on genetic algorithm	Compression Ratio	11.6	10.82	9.2	1 9.41
	PSNR	14.76	10.11	11.14	15.84
	Code Time	2.03	2.99	2.03	2.98
Fractal image compression based on scatter search algorithm	Compression Ratio	4.7	5.9	6.39	7.08
	PSNR	16.76	13.14	17.84	17.93
	Code Time	2.01	2.02	2.73	2.84

5. Conclusions

An improved scatter search algorithm is introduced to enhance the fractal image compression technique. What distinguishes the present algorithm from the original are the differences in the primary phases. The modification of the fractal image compression algorithm via the SS method is performed. By this method, various experimental solutions are generated to be modified. In each iteration, 10 solutions are generated for the image, followed by other 10 until getting the best solution or a 100 trial is achieved. After each 10 generated solution, they are compared to get the best and during this process, either the image resolution get improvement or still as it is, sometimes, we may not need more than 1 or 2 repetitions to get such improvement. The local search algorithm is used to calculate the minimum distance between their elements. The two best solutions are positioned in a hyper rectangle. However, the high-quality image with high compressed ration and less error is obtained. Besides, it helps to accelerate the encoding time, which is needed to find the appropriate IFS for each image. The analysis of the SS algorithm in comparing to two fractal image compression methods, the original Jacquin method and GA, shows its high performance among them.

References

[1] M.F. Barnesly, *Fractal everywhere*, Second edition, Academic Press, 1988.

[2] J.E. Hutchinson, *Fractals and self-similarity*, Indiana University Journal of Mathematics, 30 (1981), 713-747.

[3] M.F. Barnsley, S. Demko, *Iterated function systems and the global construction of fractals*, In Proceedings of the Royal Society of London, A399 (1985), 243-275.

[4] A.E. Jacquin, *A novel fractal block-coding technique for digital images,* in Proceedings ICASSP-90 (IEEE International Conference on Acoustics, Speech and Signal Processing), 4 (1990), (Albuquerque, NM, USA), 2225-2228.

[5] A.E. Jacquin, *Image coding based on a fractal theory of iterated contractive image transformations*, Image Proc., IEEE Trans., 1 (1992), 18-30.

[6] M.F. Barnsley, A.D. Sloan, *A better way to compress images*, Byte Mag., 1988, 215-223.

[7] M.F. Barnsley, A.D. Sloan, *Methods and apparatus for image compression by iterated function system*, U.S. Patent, 4 (1990), 941.

[8] Y. Fisher, *Fractal image compression: theory and application to digital images*, Springer Verlag, 1995.

[9] J.H.J. Han, *Fast fractal image compression using fuzzy classification*, Fifth International Conference on Fuzzy, Systems and Knowledge Discovery, 3 (2008), 272-276.

[10] Chauhan, Munesh Singh, Negi, Ashish, Rana, Prashant Singh, *Fractal image compression using dynamically pipelined GPU clusters*, Proceedings of the Second International Conference on Soft Computing for Problem Solving (Soc ProS 2012), 575-581, 2012.

[11] M. Ismail, B.M. Basha, B. Eswar Reddy, *Improved fractal image compression using range block size*, In: IEEE Proc International Conference on Graphics, Vision and Information Security (CGVIS), 2015, 284-289.

[12] N.M.G. Al-Saidi, A.H. Ali, *Towards enhancing of fractal image compression performance via block complexity*, Annual Conference on New Trends in Information, Communications, 2017.

[13] N.M.G. Al-Saidi, Sh. S. Al-Bundi, N.J. Al-Jawari, *A hybrid of fractal image coding and fractal dimension for efficient retrieval method*, Comp. Appl. Math., 2018.

[14] A.H.Y. Saad, MD. Z. Abdullah, *High-speed fractal image compression featuring deep data pipelining strategy*, IEEE ACCESS, 2018.

[15] H.Y. Abdul-Wahed, N.M.G. Al-Saidi, *Multiple-point variogram features for evolving of fractal image compression technique*, AIP Conference Proceedings, 2086 (2019), 030001.

[16] B. Varghese, S. Krishnakumar. *A novel fast fractal image compression based on reinforcement learning*, Int. J. Computational Vision and Robotics, 9 (2019), 559.

[17] W. Li, Q. Pan, S. Liang, J. Yin Jiao, *Research on fractal image compression hybrid algorithm based on convolutional neural network and gene expression programming*, Journal of Algorithms, Computational Technology, 13 (2019), 1-15.

[18] L. Thomas, F. Deravi, *Region-based fractal image compression using heuristic search*, IEEE Transactions on Image Processing, 4 (1995), 832-838.

[19] J. Levey Vehel, I. Rudomin, *Genetic algorithms for fractal image and image sequences compression*, Comptacion Visual, 1997.

[20] S. K. Mitra, C. A. Murthy, M. K. Kundu, *Technique for fractal image compression using genetic algorithm*, IEEE Transactions on Image Processing, 7 (1998), 586-593.

[21] Y. Chakrapani, K. Soundara Rajan, *Genetic algorithm applied to fractal image compression*, ARPN Journal of Engineering and Applied Sciences, 4 (2009), 53-58.

[22] N. Kulkarni, Amita, et al., *Fractal image compression using genetic algorithm with ranking select mechanism*, Communication, Information, Computing Technology (ICCICT), International Conference on IEEE, 2015

[23] Sh.S. Al-Bundi, N.M.G. Al-Saidi, N. J. Al-Jawari, *Crowding optimization method to improve fractal image compressions based iterated function systems*, (IJACSA) International Journal of Advanced Computer Science and Applications, 7 (2016).

[24] J. Li, D. Yuan, Q. Xie, C. Zhang, *Fractal image compression by ant colony algorithm*, The 9th International Conference for Young Computer Scientists, 2008, 1890-1894.

[25] C.-C.Tseng, J.-G. Hsieh,, J.-H. Jeng, *Fractal image compression using visual-based particle swarm optimization*, Image and Vision Computing, 26 (2008), 1154-1162.

[26] Y. Chakrapani, K. Soundararajan, *Implementation of fractal image compression employing particle swarm optimization*, World Journal of Modeling Simulation, 6 (2010), 40-46.

[27] N.M.G. Al-Saidi, SH. S. Al-Bundi, N.J. Al-Jawari, *An improved harmony search algorthm for reducing computational time of fractal image coding*, Journal of Theoretical, Applied Information Technology, 95.8, 2017.

[28] R. Menassel, B. Nini, T. Mekhaznia. *An improved fractal image compression using wolf pack algorithm*, Journal of Experimental, Theoretical Artificial Intelligence, 2017.

[29] X. Sun, W. Sun, J. Wang, Y. Zhang, Y. Gao, , *Using a Grey–Markov model optimized by Cuckoo search algorithm to forecast the annual foreign tourist arrivals to China*, Tour. Manage, 52 (2016), 369-379.

[30] B.M. Ismail, B.E. Reddy b, T.B. Reddy, *Cuckoo inspired fast search algorithm for fractal image encoding*, Journal of King Saud University–Computer and Information Sciences, 30 (2018), 462-469.

[31] F. Glover, *Heuristics for integer programming using surrogate constraints*, Decision Sciences, 8 (1977), 156-166.

[32] F. Glover, *Scatter search and path relinking: foundations and advanced designs*, New Ideas in Optimization, 1999, 297-316.

[33] F. Glover, *Tabu search for nonlinear and parametric optimization (with links to genetic algorithms),* Discrete Applied Mathematics, 49 (1994b), 231-255.

[34] F. Glover, M. Laguna, *Tabu search*, Kluwer Academic Publishers, Boston, 1997.

[35] F. Glover, M. Laguna, R. Martí, *Fundamentals of scatter search and path relinking*, Control Cybern., 29 (2000), 652-684.

Accepted: June 04, 2020