Diversity in Crowding Selection to Improve Fractal Coding Technique Performance

التنوع في اختيار الازدحام لتحسين أداء تقنية التشفير الكسورية

Naseif J. AL-Jawari[1], Nadia M. G. Al-Saidi[2], and Shaimaa S. Al-Bundi[3]
[1]Department of Mathematics, College of Sciences, Mustansiriyah University, Iraq
[2]Applied Sciences Department, University of Technology, Iraq
[3]Department of Mathematics, College of Education for Pure Sciences, Ibn Al-Haitham-Baghdad University, Iraq
E-mails: [1]njaljawari@hotmail.com, [2]nadiamg08@gmail.com, and [3]shaimaaalbundi@gmail.com

Abstract

Since Barnsley conception of fractal coding, this research topic was rapidly grown and considered as the most promising technique. The high encoding time prevents the popularity of this technique. Crowding as an improved Genetic algorithm (GA) is used for fractal coding method to improve the searching computational time between range- domain blocks. Although, crowding method is one of the approaches that maintain population diversity, but still the offspring replaces the most similar individual. In this method, each offspring is completed with one of its parents and replaced it. That means, crowding method concentrated on best element's population search space, which makes the new generation influenced by the properties of the previous generation. To support global exploration and prevent trapping in local optima, a new probability is added, which helps to satisfy diversity in the population selection from both the best and worst individuals. This prevents the algorithm from being trapped in local optima. By this method, the time complexity of the algorithm is improved because it makes a balance between reaching the optimal solution and population diversity.

المستخلص

منذ ان طرح العالم Barnsley مفهوم ترميز الكسور، تطور هذا المفهوم تطورا سريعا وأعتبر التقنية الواعدة لكن وقت الترميز لهذه التقنية كان حائلا دون رواجها. تقنية الازدحام طورت النظرية الجينية باستخدام طريقة الترميز الكسورية فقد حسنت وقت حساب الترميز بين كتل المجموعة-النطاق. وعلى الرغم من ان طريقة الازدحام واحدة من الطرق التي تحافظ على التنوع السكاني، لكن لا تزال وليدة محل الفرد الأكثر امثلية. في طريقة الازدحام، بعد الانتهاء من كل جيل يقوم بالاكمال مع أحد الوالدين ويحل محل احدهما. وهذا يعني، طريقة الازدحام ركزت على أفضل عناصر السكان في فضاء البحث، مما يجعل الجيل الجديد تتأثر بخصائص الجيل السابق. ولذلك ولدعم البحث العام وتجنب الوقوع في المنطقة ذات الامثلية المحلية، يتم اضافة احتمال جديد، يساعد على تلبية التنوع في اختيار السكان من المجتمعين الأسوأ والافضل. هذا يمنع الخوارزمية من الوقوع في الامثلية المحلية. بهذه الطريقة المقترحة، يتم تحسين التعقيد والتأخير في وقت الخوارزمية لأنها تعمل موازنة بين الوصول إلى الحل الامثل وتنوع السكان.

1. Introduction

Using iterated function system (IFS) to encode real-world images is considered as an innovative technique in the field of image compression. By this technique, the image is reconstructed implicitly as a fixed point of the image operator after some iteration. Therefore, finding suitable class of image operators is still a challenges that motivate many researchers to continue work in this attractive research topic in order to design the best algorithms for encoding and decoding of images, taking into consideration satisfying tradeoff between the three basic factors in any compression system which are; compression ratio and PSNR (the quality measure) and computational complexity for the encoding and decoding process.

Mandelbrot [1] was the first who introduced fractal concepts. Expressing an image as an attractor of an iterative process for the set of maps was introduced by Barnsley [2]. This set that generates the attractor was called iterative function system (IFS). It can be found by collage theorem that states "to find an IFS whose attractor looks like a given image, a set of transformations which are contraction mappings of the image should be found".

This new vision motivates the researchers to focus on using fractal concepts for image compression, which help to represent an image as coefficients of a set of affine transformation. This approach failed to represent the given image as a single collage (i.e representing of the given image by a single set of affine transformation). This defect leads Jacquin (Barnsley's Ph.D. student) to introduce the idea of partitioned IFS [3]. By this idea, the image is partitioned into blocks and each block is handled separately as an attractor of an IFS. In spite of the feasibility of this technique with all of its advantages, still, there is an intensive computation needed in the encoding process.

Since the most computation for generating of IFS coefficients is spent for a finding of the best match between image parts, therefore, using an approach that has been proven to be very helpful in the searching process is of great demand. Genetic Algorithm (GA) has been proven to be an important approach for obtaining a solution to complicate search problems. Many studies for using GA to improve the searching time in fractal coding technique have been proposed [FIC in GA]. This approach is improved by De Jong [4] when he introduced crowding method to maintain population variety and speed convergence. This new method is improved and applied for the first time in fractal coding method by Al- Bundi et al. [5]. With this method, the possibility of trapping in local optima is still may happen. Therefore, in this work, a new probability is added to satisfy the diversity in the selection phrase. The new selection includes both the best and worst individuals. By this improvement, we can avoid trapping in local optima.

In addition to this introductory section, the rest of this paper is organized as follows: Section 2 presents some mathematical background for fractal theory. The methods of genetic and crowding are presented in Section 3. The proposed method is presenting in section 4. The implementation and analysis is discussed in Section 5. Section 6 concludes the paper.

2. Theoretical Background for Iterated Function System

In this section, we presented the principles that are considered as fundamental in fractal theory. The reader interesting in more details can be referred to see [2,6].

Definition 1 [2]: Let *(X, d)* be a metric space, a sequence (X_n) is said to be Cauchy sequence, if for any given $\varepsilon > 0$, we have $d(X_m, X_n) < \epsilon$ for all *m, n* $\in N$ (natural numbers). *(X, d)* is called complete if every Cauchy sequence in *X* converges to an element of *X*.

Definition 2 [2]: Let $f: R^2 \rightarrow R^2$ be a transformation of the form $w(x_1, x_2) = (ax_1 + bx_2 + e, cx_1 + dx_2 + f)$ where a, b, c, d, e and f are real numbers. This transformation is called a (two dimensional) affine transformation.

$$w(x) = w_i \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \quad .... (1)$$

The following equivalent notations have been used:

where $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is 2×2 real matrix and *t* is the column vector $\begin{pmatrix} e \\ f \end{pmatrix}$, where $(e,\ f) \in\ R^2$.

The matrix *A* can always be written as follows:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} r_1 \cos\theta_1 & -r_2 \sin\theta_2 \\ r_1 \sin\theta_1 & r_2 \cos\theta_2 \end{pmatrix} \quad .... (2)$$

Definition 3 [2]: Let $f: X \rightarrow X$ be a transformation on a metric space (X, d). It is called contractive or contraction mapping if there is a constant $0 \leq s < 1$ such that $d(f(x), f(y)) \leq s.d(x, y)$ for all $x, y \in X$, any such number *s* is called contractivity factor of f.

Definition 4 [2]: Let $f: X \to X$ be a transformation on a metric space *(X, d)*, a point $x_f \in X$, such that $f(x_f) = x_f$ is called fixed point of the transformation f.
The fixed point is very important; it represents which part of the shape that we are interested in is not affected by the transformation (fixed).
One of an important concept in the fractal theory is the Hausdorff metric. Therefore, many mathematicians have discussed and proved the basic concepts and results of the Hausdorff metric [7-8]. It is known as the space of fractals and denoted by $H(X)$, and generated from the complete metric space X by taking all the compact sets in X. The important distance defined on $H(X)$ is known as $h(d)$ and defined as follows.
Definition 5 [6]: Let *(X, d)* be a complete metric space. The Hausdorff distance h can be defined between any two points A and B in $H(X)$ as,

$$H(A,B) = \max\{sup_{a\in A}\, h(a,B), sup_{b\in B}\, h(b,A)\} \qquad \text{.... (3)}$$

where $h(a,B) = \inf\{h(a,b), b \in B\}$ is the distance from a point a to the set B.

Definition 6 [2]: Let (X, d) be a complete metric space. An *IFS* on X is a finite set of contractive self-mappings $w_i: X \to X$ with respective contractivity factors s_i for $i = 1, 2, \dots, N$, such that, $s = \max\{s_i, i = 1, 2, \dots, N\}$. *IFS* is based on the affine transformation $w(x)$, where,

$$w(x) = w\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_1 \cos\theta_1 & -r_2 \sin\theta_2 \\ r_1 \sin\theta_1 & r_2 \cos\theta_2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} \qquad \text{.... (4)}$$

Definition 7 [2]: In *IFS*, any compact subset (fixed point) $A \in H(X)$ is called attractor for *IFS* if,

$$A = \bigcup_{n=1}^{n} w_n(A) \qquad \text{.... (5)}$$

The fixed point observes existence and uniqueness based on the contraction mapping theorem. An important property of any *IFS* is the attractor, that is mean, the attractor A is fully known by the set of coefficients of W, and can be generated by applying the *IFS* to any starting point several times.

Definition 8 [6]: *IFS* theory is an important object in fractals field and supplied powerful tools for the search of fractal. They are applied for both generating, and modelling of the irregular patterns. It can be viewed as an image compression when the latter is associated to an automatic generation of fractal forms and irregular forms. In both of these applications, the generated information is called attractor of the IFS. The work on the system of the contractive mappings to make fractal sets has been considered by many researchers and most fractal image compression methods are based on *IFS*.
IFS is developed by Hutchison (1981) [6], Barnsley and others (1985, 1986, 1988) [8-9]. These systems of mapping were discussed widely and used in many applications, such as image compression method. The general formula for *IFS* is introduced in the following definition:

Definition 9 [2]: Let (X, d) be a complete metric space. An *IFS* on X is a finite set of contractive self-mappings $w_i: X \to X$ with respective contractivity factors s_i for $i = 1, 2, \dots, N$ such that $s = \max\{s_i, i = 1, 2, \dots, N\}$.
IFS is based on the affine transformation given by:-

$$w(x) = w\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} r_1 \cos\theta_1 & -r_2 \sin\theta_2 \\ r_1 \sin\theta_1 & r_2 \cos\theta_2 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

Sometimes, the world (hyperbolic) is used with the iterated function systems since it sometimes drops in practices. The following theorem will show why we use the concept of hyperbolic *IFS*.

Theorem 1 [2]: Let (X, d) be a metric space, let $\{w_1, w_2, \dots w_N\}$ be a hyperbolic iterated function system where $w_i, i = 1, \dots, N$ is a contraction mapping. Let $W: H(X) \to H(X)$ defined by

$W(B) = \bigcup_{n=1}^{n} w_n(B)$ …. (6)

Then W is a contraction mapping with contractivity factor s on the complete metric space $(H(X), h(d))$. Therefore,

$h(W(B), W(C)) \leq s \cdot h(B, C)$ …. (7)

for all $B, C \in H(X)$. Let $A \in H(X)$ be defined as;

$A = W(A) = W^n(A)$ …. (8)

Then A is a unique fixed point given by;

$A = \lim_{n\to\infty} W^n(B)$ …. (9)

for any $B \in H(X)$

Definition 10 [2]: In *IFS*, any compact subset (fixed point) $A \in H(X)$ is called attractor for *IFS* if;

$A = \bigcup_{n=1}^{n} w_n(A)$ …. (10)

The fixed point observes existence and uniqueness by the contraction mapping theorem. An important property of any *IFS* is the attractor, that means the attractor A is fully known by the set of coefficients of W, and can be generated by applying the *IFS* to any starting point several times.

3. Fractal Image Compression

The fractal image compression (FIC) approach is an important search area with many possible application fields. This approach is focused on finding fractal code that generates given objects. The major problem of standard fractal image coding is its time consuming compared with another image coding methods. Some time is spent in searching for a similar domain block. Therefore, new techniques to solve this problem and accelerate this method are in great demand.

3.1 Jacquin Approach for Fractal Image Coding Technique

Fractal image coding makes good uses from self-similar images in space by reducing image geometric redundant. Fractal coding operation is very complicated but decoding operator is quite simple, that makes use of possibility in a high compression ratio. The principle theory of the fractal image coding scheme is depended on iterated function system attractor. Now getting partition iterated function system (PIFS) parameter is regarded as the main problem in fractal coding [10]. In each search for similarity between the range and domain blocks, two types of blocks are emerged from the same image, as shown in Figure 1.

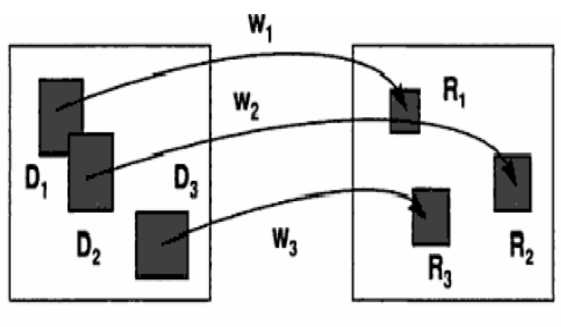
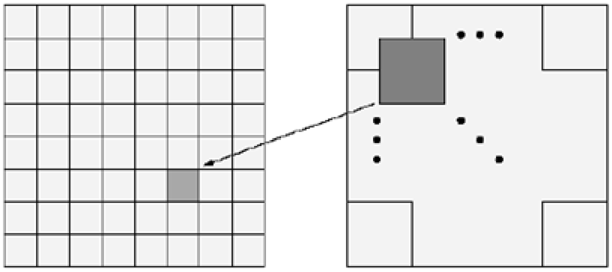


Figure 1: Clarification of PIFS

In the following, is an explanation of the main processes in Jacquin technique for fractal image coding.

1. A given image M is partitioned into non-overlapping blocks $R=\{R_1, R_2, \dots, R_m\}$ of size $r \times r$ called range blocks, where $m = (M/r)^2$, and overlapping blocks $D=\{D_1, D_2, \dots, D_n\}$ of size $2r \times 2r$ called domain blocks, where $n = (M - 2r + 1)^2$ as shown in Figure 2 (a,b). As an example, for M of

size 128×128 if the size of each range block is 8×8, then we have $(128/8)^2 = 256$ range blocks and $(128\text{-}2\times8+1)^2 = 12{,}769$ domain blocks of size 16×16.



a- Range Blocks b- Domain block

Figure 2(a,b): Domain and range blocks of the PIFS

2. For each range block R_i, choose its approximate domain D_k, $k=1,\ldots,n$ and a suitable contractive affine transformation w_{ik} that satisfied (3.1), which is given by:

$$d\left(R_i, w_{ik}(D_k)\right) = min\, d\left(R_i, w_{ij}(D_j)\right) \qquad \ldots.\ (11)$$

where w_{ik} is the contractive affine transformation from D_j to R_i. This can be represented by $d\left(R_i, w_{ij}(D_j)\right)$ where d refers to the mean square error (MSE) between R_i and w_{ij} that consist of two mappings T_j and θ_{ij} such that:

$$w_{ij} = \theta_{ij} * T_j$$

Where T_j represented the contraction transformation that transformed D_i size into R_i size. This transformation is described as follows:

The domain block D_j is divided into non-overlapping blocks, if they for example of size 2×2, the pixel value of the transformed block $T_j(D_j)$ is computed from the average of the four pixels in the block of D_j, as shown in Figure 3.

Figure 3 The contraction of a domain block.

The mapping θ_{ij} is performed in two steps; firstly, the block $T_j(D_j)$ is transformed using eight isometries given in Figure 4.

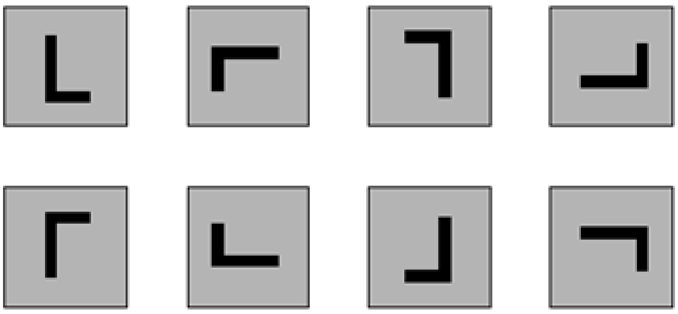


Figure 4: The eight isometries.

Second step, the pixel values of the resulting block from first step is transformed by W_{ij}, where W_{ij} is defined as follows:

$$W_{ij}(z) = s_i z + o_i \qquad \text{....(12)}$$

Here z represents the pixel value that is obtained from the first step and the scaling s_i and the offset o_i defined in (13) and (14) respectively are calculated by the mean square error (*MSE*) of the pixel values of the range block R_i.

$$s = \frac{n^2 \sum_{i=1}^{n^2} a_i\, b_i - \left(\sum_{i=1}^{n^2} a_i\right)\left(\sum_{i=1}^{n^2} b_i\right)}{n^2 \sum_{i=1}^{n^2} a_i^2 - \left(\sum_{i=1}^{n^2} a_i\right)^2} \qquad \text{...(13)}$$

and $$o = \frac{1}{n^2}\left[\sum_{i=1}^{n^2} b_i - s \sum_{i=1}^{n^2} a_i\right] \qquad \text{...(14)}$$

The *MSE* difference is calculated using:

$$RMS = \frac{1}{n}\left[\sum_{i=1}^{n^2} b_i^2 + s\left(s.\sum_{i=1}^{n^2} a_i^2 - 2\sum_{i=1}^{n^2} a_i b_i + 2.o\sum_{i=1}^{n^2} a_i\right) + o.\left(o.n^2 - 2\sum_{i=1}^{n^2} b_i\right)\right] \quad \text{...(15)}$$

This representation contains two parts, the massive part to represent the image's pixels and the geometric part to represent the isometric and scaling.

4. Genetic and crowding approach

A. Genetic Algorithm for Fractal Image Compression

Searching processes begin after dividing the image into range and domain blocks as in standard Jacquine approach image compression. For each range block, the domain block and identical transformations that best cover the range block is specific. In general, for best correspondence, transformation codes are set, including contrast and brightness. Searching succeeds when the domain block is appropriate for the suitable range block. Eventually, mapping data are stored. Then, we use GAs to attain the intended outcome via fractal compression. Usually, GAs are employed to find the near optimal solution, and thus the GA for fractal compression of images is shown below [11].

1- Chromosomes

Given that, the GA works on the chromosomes, producing chromosomes from the range and domain blocks is a crucial step in using the GA for *FIC*. The transformation parameters obtained for each block are coded on a set of a fixed number of bits. These parameters are then stored as chromosomes. By encoding the parameters of an image, a chromosome comprises *N* genes that are equal to the number of the non-coded parts of an image. These genes are generated from parameters X_D and Y_D, which refer to the coordinates of the domain block, and the flip refers to the transformation isometrics. Figure 5 shows the chromosomes

<table>
<tr><th colspan="10">Range Block</th></tr>
<tr><td colspan="3">Block1</td><td colspan="3">Block2</td><td>.........</td><td colspan="3">Block N</td></tr>
<tr><td>X_D^1</td><td>Y_D^1</td><td>$Flip^1$</td><td>X_D^2</td><td>Y_D^2</td><td>$Flip^2$</td><td>.........</td><td>X_D^N</td><td>Y_D^N</td><td>$Flip^N$</td></tr>
</table>

Figure 5: Image representing a chromosome

2- Fitness Function

Fitness function is a specific task for each chromosome, which refers to the capability of each chromosome to survive and proliferate. We denote fitness as the value of error between the coded range and domain blocks that are assigned by the transformation with analogous luminance and contrasting values. The error is computed using the root mean square equation (15).

3- Genetic Operators

Crossover and mutation are two basic operators that are used in all implementations of genetic algorithms. These operators are described as follows.

- *Crossover Operator***:** The crossover operator selects two parents based on their fitness, and then attempts to produce a new child with the best possible quality. A high fitness value provides the crossover operator with a high probability of selection. The crossover operator changes the genes of the parent. Given that a random number a is produced in the interval [0, 1], the new coordinates are computed using the equation below.

First offspring $X_D = a * X_{D1} + (1-a) * X_{D2}$

$$Y_D = a * Y_{D1} + (1-a) * Y_{D2} \qquad \ldots(16)$$

Second offspring $X_D = (1-a) * X_{D1} + (1-a) * X_{D2}$

$$Y_D = (1-a) * Y_{D1} + (1-a) * Y_{D2}$$

- *Mutation Operator*: The mutation operator changes the value of one or more genes in the chromosome, thereby adding completely new gene values to the gene pool. The GA may achieve a better solution using these new values. The mutation operator also introduces the verity in the chromosomes. The information changes randomly based on the mutation rate.

***B.* Crowding Method for Fractal Image Compression**

The crowding method [12] is proposed to eliminate the selection process and introduce a preselecting process. This will cause a very fast GA to be used for multidimensional optimization problem. By reducing the selection process, the individuals are mutated randomly with any other population individuals. During the replacement process, the paining between the offspring and one of the parents is performed first. This operation is done with probability P_c. This pairing process is happened according to the similarity between them. In the evaluation step, the fitness function which represented the least square error between the offspring and the parents is responsible for deciding about which individual of the population is allowed to stay.

This section proposed an improved crowding method in order to be applied to improve *FIC*. With this method, the diversity is preserved in the population with the opportunity for each individual to be a parent. What distinguishes our proposed method from the original crowding method [12] and Mahfoud method [13] is some technical differences in the main phases of the algorithm. The population set $\{T_1, T_2, \ldots, T_n\}$ is constructed by finding all contruction mapping T_i that resulted from the similarity measure between the range block and domain block of the query image. This set is calculated using Jacquine approach [14]. Each individual T_i is assigned a fitness value $f(T_i)$ as its weight, where $f_i \in \{f_1, f_2, \ldots, f_n\}$ represents the minimum distance between R_i and D_j, j=1,…,m. This value is used in the selection process of the parents $\{P_1, P_2\}$ and controlled by a chosen factor known as crowding factor that determines the number of the maximum selection of this individual as shown in Figure 6.

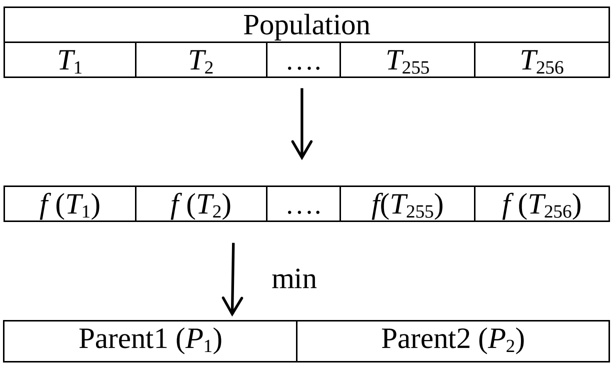


Figure 6: Selection process

5. The proposed diversity method

Since the genetic algorithm is based on the natural independent for reproduction which are the selection, crossover and mutation, therefore, controlling of these operations is needed to achieve a balanced and ensure best new generation. Hence, population diversity is great demand. This section proposed a new method to maintain the population diversity. This method is an improvement to the crowding method proposed in section (4).

Although, crowding method is one of the approaches that maintain population diversity, but still the offspring replaces the most similar individual. In this method, each offspring is completed with one of its parents and replace it. That means, crowding method concentrated on best elements population search space which makes the new generation influenced by the properties of the previous generation. To support global exploration and prevent trapping in local optima a new probability is added, which helps to satisfy diversity in the population selection from both the best and worst individuals. This will reduce the algorithm from being trapped in local optima. This method will improve the time complexity of the algorithm because it makes a balance between reaching the optimal solution and population diversity.

```
Diversity Method
Input: Image I, P_d              % P_d is known as a crowding factor
p_c                              % crossover probability
p_m                              % mutation probability
Output: New optimized population cp = {T'_1, T'_2, ..., T'_n}
Bagin
    Genetic diversity population cp = {T_1, T_2, ..., T_n}
    Evaluate (cp)                % calculate the fitness for cp
  while terminated () do
    P = selection (cp, P_d)      % select parents P = {T: T ∈ TP,
                                   such that, RMS (T(i), i) < ε}
    Offspring = recombination (cp, p_c, P)
    Offspring = mutation (offspring, p_m)
    Compare offspring with parents and add the best one to the cp
   end while
end.
```

In the selection phase, the parents are chosen according to the minimum fitness values, where the maximum number of selection of each individual equals P_d (diversity factor).

```
Selection function
Input:   c_p                     % crowding population
S_p                              %Select probability
P_d                              % maximum number the selection
Output:  P                       % parents
Begin
S_p = random number
for i = 1 to n                   % n is the size of c_p
    p_i = f_i / Σ_{j=1}^{n} f_j  % f is the fitness
end
while z = false do
If S_p >= 0.7 then
    Select the best two individual P from cp according to p_i
            and call it P
    If p_c f <= P_d then
```

Z = true
Else
Select the worst two individual P from cp according to p_i
and call it P
If $p_c\ f$ <= P_d then
Z = true
end if
end while
end

In the recombination phase, the offspring is generated according to two logical values as given in the recombination algorithm in proposed crowding method

After the recombination phase, the resulting offspring are competed with the mutation phase for surviving of their parents. The decision of winning is taken based fitness value (the similarity measure between the offspring and the parents) in order to decide the one that should appear in the new population, such that:

If *RMS* (*P*,*C*) < ε then *C* is the winner of the competition

Else *P* is the winner of the competition

The mutation algorithm for diversity as in the proposed crowding method. The termination value of the algorithm is deduced according to learning process on a sample of different images to determine the best that can satisfy the compromising between the optimum solution and the execution time.

6. Implementation and Analysis

This method is also tested on different range sizes, which are: 2,4 and 8. It is found that the quality of the decoded image for range size 2×2 and 4×4 is better than the one with range size equal 8×8. Table 3.5 illustrates the reconstructed image after applying the decoding algorithm for different range size. Table (1) illustrates coding-decoding, time and compression ratio on the selection images using the suggested technique. Table (2) illustrates the peak signal-to-noise ratio (PSNR) and MSE on the selection images using the suggested technique (diversity method).

Table 1: Coding, decoding time and compression ratio based on the diversity method for different range size

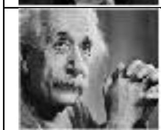
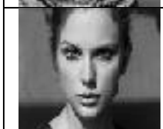
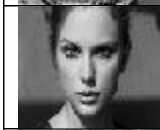
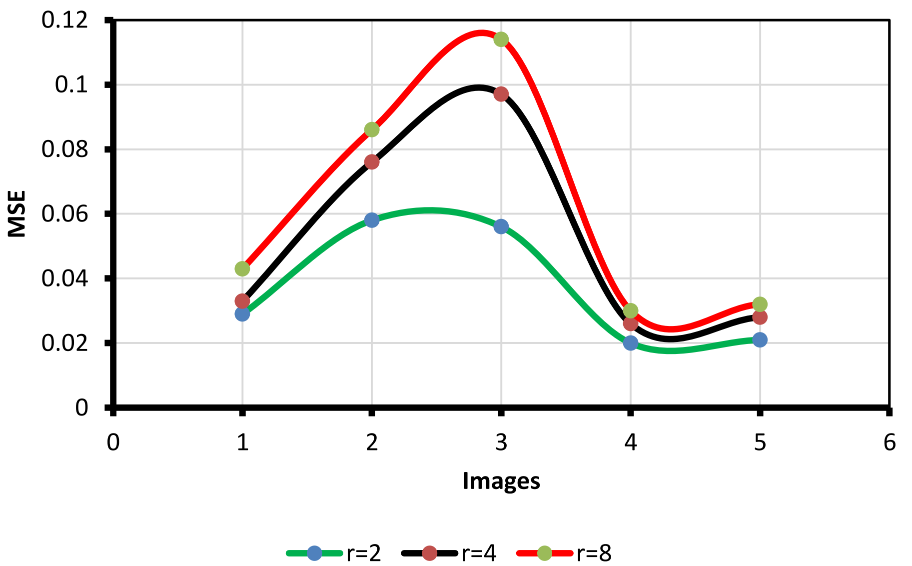
Images	Range	Coding Time	Decoding Time	Compression Ratio
	2×2	7.696 sec	0.586 sec	3.25
	4×4	2.03 sec	0.20 sec	11.6
	8×8	0.247 sec	0.176 sec	9.97
	2×2	7.693 sec	0.585 sec	2.48
	4×4	2.03 sec	0.20 sec	9.21
	8×8	0.227 sec	0.175 sec	7.53
	2×2	8.190 sec	0.564 sec	3.63
	4×4	2.99 sec	0.20 sec	10.82
	8×8	0.226 sec	0.1765 sec	7.33
	2×2	7.769 sec	0.564 sec	2.35
	4×4	2.98 sec	0.20 sec	9.41
	8×8	0.221 sec	0.176 sec	11.16
	2×2	7.779 sec	0.557 sec	3.23
	4×4	2.97 sec	0.20 sec	12.1
	8×8	0.219 sec	0.177 sec	12.31

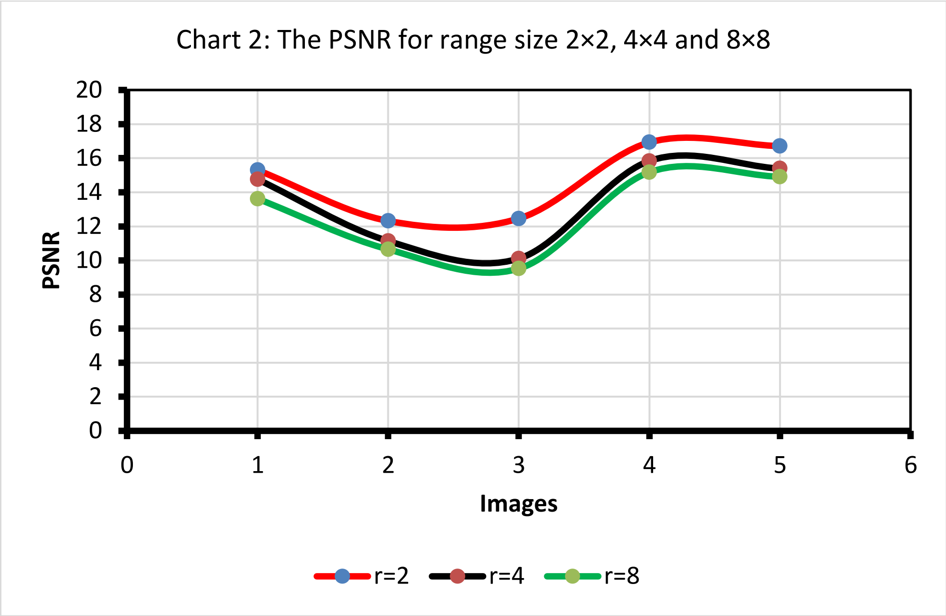
Table 2: PSNR and MSE for different range block

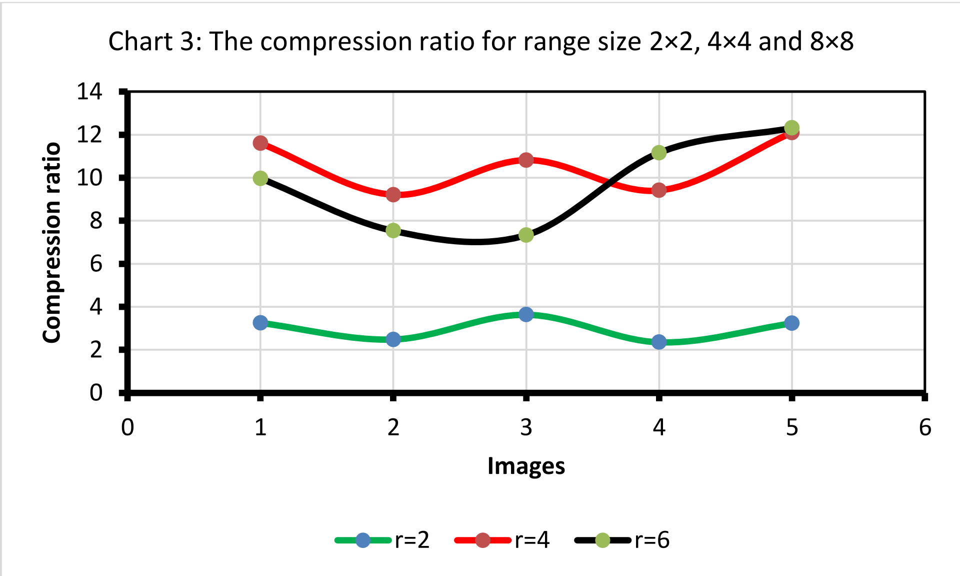
Images	Range	PSNR	MSE
	2×2	15.30	0.029
	4×4	14.76	0.033
	8×8	13.62	0.043
	2×2	12.32	0.058
	4×4	11.14	0.076
	8×8	10.65	0.086
	2×2	12.46	0.056
	4×4	10.11	0.097
	8×8	9.52	0.114
	2×2	16.93	0.020
	4×4	15.84	0.026
	8×8	15.18	0.030
	2×2	16.71	0.021
	4×4	15.39	0.028
	8×8	14.91	0.032

The results of Tables 1 and 2 are illustrated in the following charts

Chart 1: The MSE for range size 2×2, 4×4 and 8×8







7. Conclusions

An improved crowding method is proposed in order to be applied to improve FIC. The proposed method is preserved the population with the chance for each individual to be a parent. What distinguishes the present proposed method from the original are some technical differences in the main phases of the algorithm. A more modification of crowding method is performed by adding select probability as a random number to algorithm of crowding method. This diversity in the selection process helps to get a higher quality image with less time and MSE.

References

1. B. B. Mandelbort, "Fractal Geometry of Nature", San Francisco: Freeman, 1982.
2. M. F. Barnesly, "Fractal Everywhere", Second edition, Academic Press, 1988.
3. A. E. Jacquin, "A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding", PhD., Georgia Institute of Technology, 1989.
4. K. A. De Jong, “An Analysis of the Behavior of a Class of Genetic Adaptive Systems”, PhD thesis, Department of Computer and Communication Sciences, University of Michigan, Ann Arbor, MI, 1975.
5. S.S.Al-Bundi, N.M. Al-Saidi, N. J. Al-Jawari, “Crowding Optimization Method to Improve Fractal Image Compressions Based Iterated Function Systems”, International Journal of Advanced Computer Science and Applications, Vol. 7, No. 7, 2016.
6. J.E. Hutchinson, Fractals and self-similarity, Indiana University journal of mathematics, 30(5), 713-747, 1981.
7. M.F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In Proceedings of the Royal Society of London A399, 243 275, 1985.
8. E. Jacquin, "Image coding based on fractal theory of iterated contractive image transformations", IEEE Trans. on Image Processing, 1, 18-30, January 1992.
9. B. Mandelbort, "How long is the coast of Britain? Statistical self-similarity and Fractional dimension", science 156 (3775), pp: 636 – 638, 1967.
10. A. E. Jacquin, "A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding", PhD., Georgia Institute of Technology, 1989.
11. Holland, J.H.: Adaptation in natural and artificial systems. The University o Michigan Press, 1975.
12. O. J. Mengshoel, D. E. Goldberg, “The Crowding Approach to Niching in Genetic Algorithms”, Evolutionary Computation”, Vol. 16, No. 3, pp. 315-354, 2008.
13. S. W. Mahfoud, “Niching Methods for Genetic Algorithms”, PhD thesis, Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 1995.

M. F. Barnsley, A. D. Sloan, “A Better Way to Compress Images”, Byte Mag., pp. 215–223, 1988.