Comparison Between New Operators Of Genetic Algorithm To Implementation Face Image

Shahla H. Karruffa
Basic Sciences , College Of Dentistry , Mosul University , Mosul , Iraq
(**Received:** 11 / 3 / 2009 ---- **Accepted:** 25 / 10 / 2009)

Abstract
Genetic algorithm (GA) are adaptive, robust, efficient, and global search methods, suitable in situations where the search space is large. They optimize a fitness function, corresponding to the preference criterion to arrive at an optimal solution using certain genetic operators.
GA are able to produce valuable information for an enterprise (Like Face images), it is only when the GA are implemented thoroughly and by a resource qualified to utilize the data that an enterprise can fully capitalize on the benefits.
I have dealt with in this research with basic processes in the GA like selection, crossover and mutation. The concentration was on mutation being has an important on the results. New five methods were suggested for mutation process for implementing face image, and these methods are: (1- Separation genetic operator, 2- Annealing genetic operator, 3- Swap genetic operator, 4- Block-swap genetic operator, 5- Suffle genetic operator).
Later, the comparing and finding the best method among them, where the research includes dividing the work into three stages. In the first stage when implementation number is low, while the second one when implementation number is intermediate ,but the third stage when implementation number is high.
The program was done in Matlab language (Version 6.5) for performing the work and dealing with each image alone and the images were displayed and drawing the diagram sketches for the resulted image.

1- Genetic Algorithm (GA):
Genetic Algorithms (GA) are computing algorithms constructed in analogy with the process of evolution. [1] [2] GA seem to be useful for searching very general spaces and poorly defined spaces. [3] [4].
GA is one of the stochastic optimization methods, which is simulating the process of natural evolution. [5] GA first was presented by John Holland as academic research. [6] [7] However, today, GA turned out to be one of the most promising approaches for dealing with complex systems, which at first nobody could imagine from a relative modest technique. GA is applicable to multi-objective optimization and can handle conflicts among objectives. Therefore, it is robust where multiple solutions exist. In addition, it is highly efficient and it is easy to use. [8]
GA are computational methods inspired by Darwinian evolution theory. [6] The variables are coded into the vector which is called a chromosome, an initial population of chromosomes is generated randomly. [9] The evolution is performed in an iterative manner where in each step the fitness of chromosomes is evaluated and the population is altered by the operations of selection, crossover and mutation. Each chromosome can be thought of as a point in the search space of the candidate solutions. [10] [11] the GA processes populations of chromosomes, successively replacing one population by another. The GA most often requires a fitness function that assigns a score (fitness) to each chromosome in the population. The selection selects chromosomes for reproduction according to their fitness .In crossover, the operator randomly chooses a locus in the chromosome and exchanges it between two chromosomes to create two offspring. [12] [13] Mutation randomly flips some of the bits in a chromosome. The procedure iterates until the desired fitness is reached or a predefined number of iterations are reached. As the GA is a stochastic process where the initial population is randomly created , and the other operations are also random, usually several runs are done for the same task and the results are evaluated. [14] [15] [16]
In order for a GA to efficiently search such large spaces, one must give careful thought to both the representation chosen and the evaluation function. [17] GA is being used with very high degree of success in all kinds of numerical and combinatorial optimization problems. [18] This capability of GA is due to its inherent parallelism, which when used fully, enables search in complex landscapes and to optimize various objective functions of interests in various fields of applications. [19]
In this research, a new and an efficient operators of GA is used to perform the implementation of face image.

2- Face Image:
The human face is an extremely complex object, with both rigid and non-rigid components that vary over time, some times quite rapidly. The object is covered with skin, a non uniformly textured material that is difficult to model either geometrically or photometrically. Skin can change color quickly when one is embarrassed or becomes warrn or cold. The reflectance properties of the skin can also change rater quickly, as perspiration level changes, The face is highly deformable, and facial expressions reveal a wide variety of possible configurations and changes in skin color and texture caused by exposure to sun. [20] [21]

3- Logical Crossover:

The logical crossover considers four cases [22] [23] [24]:

1-Health of father and mother greater than the average health of the entire population , In this case, the generation of two children (let us assume for simplicity the case of two children and a single crossover point) occurs in a traditional manner:

$$F_F > \bar{F} \text{ and } F_M > \bar{F} : \quad \underbrace{\begin{matrix} A'+A'' \\ B'+B'' \end{matrix}}_{\text{Parents}} \Longrightarrow \underbrace{\begin{matrix} A'+B'' \\ B'+A'' \end{matrix}}_{\text{Children}}$$

Where:

F_F : is represent (the intensity value of face image) fitness of father

F_M: is represent (the intensity value of face image) fitness of mother

$\bar{F}$: is represent the average health of the entire population.

In this research, this case is represented of crossover operators.

2-Health of both parents lower than the average health of the entire Population, In this case, the generation of two children occurs through the rejection (~) of parent's genes:

$$F_F > \bar{F} \text{ and } F_M > \bar{F} : \quad \underbrace{\begin{matrix} A'+A'' \\ B'+B'' \end{matrix}}_{\text{Parents}} \Longrightarrow \underbrace{\begin{matrix} \sim A'+\sim B'' \\ \sim B'+\sim A'' \end{matrix}}_{\text{Children}}$$

Where:

F_F : is represent (the intensity value of face image) fitness of father

F_M: is represent (the intensity value of face image) fitness of mother

$\bar{F}$: is represent the average health of the entire population.

3- 4-Health of one of the parents less than the average health and That of the other greater than the average health of the entire population, In the third and fourth cases, only the parent whose health is greater than the average health transmits his/her genes, while those of the other parent are rejected:

$$(3) F_F > \bar{F} \text{ and } F_M > \bar{F} : \quad \underbrace{\begin{matrix} A'+A'' \\ B'+B'' \end{matrix}}_{\text{Parents}} \Longrightarrow \underbrace{\begin{matrix} A'+\sim B'' \\ \sim B'+A'' \end{matrix}}_{\text{Children}}$$

Or

	Parents		Children
(4)$F_F > \bar{F}$ and $F_M > \bar{F}$:	A’+A’’	$\Longrightarrow$	~A’+B”
	B’+B’’		B’+~A”

Where:
FF : is represent (the intensity value of face image) fitness of father
FM: is represent (the intensity value of face image) fitness of mother
$\bar{F}$: is represent the average health of the entire population.

4- Mutation:
Mutations are one of the basic elements of adaptation, hence evolution. They help living beings to survive. [14] [15] [25]

	Parents		
$F_F > \bar{F}$ and $F_M > \bar{F}$:	A’+A’’	$\Longrightarrow$	~A’+~B”
	B’+B’’		~B’+~A”

Mutation is a sudden change in the chromosome in the reproduction process. It simply means a negation of a bit in the sequence assuming a bit string implementation of chromosomes. [12] [18] [26] [27] The number of mutations is again implementation dependent. This number, in common, is not evaluated directly but the mutations occur rather probabilistically. [28]
Mutation is the basic operation for Baldwin Effect. It is still the most enthusiastic part of the genetic algorithms. Mutation is another important operator which is used for fine tuning the solution. [22]
this research suggestion five methods of mutation to implementation the face image such as:

1- Separation: The separation genetic operator separates two contiguous randomly specified genes (called the first and second contiguous genes), swapping them, in inverse order, with two other randomly specified genes (called the first and second extracted genes), provided that these other two genes both precede or both follow the contiguous genes and have a distance not greater than NG/2 gene positions apart.

2- Annealing: The annealing genetic operator randomizes the succession order of the genes included in a randomly specified circular interval, but with a fixed length L (typically L = 5).

3- Swap: The swap genetic operator swaps a randomly specified gene with the following gene, and a final local optimization is performed.

4- Block-Swap: The Block-Swap genetic operator divides the individual into three randomly specified pieces and swaps the inner parts of the first two, provided that the inner parts of the three blocks have at least three genes, and a final optimization is performed.

***5- Shuffle*:** The Shuffle genetic operator divides the individual into four randomly specified pieces (A, B, C, D) and shuffles them in a specific way (A, B, C, D $\longrightarrow$ B, D, C, A), and a final optimization is performed. The suggestion methods shown in figure (1).

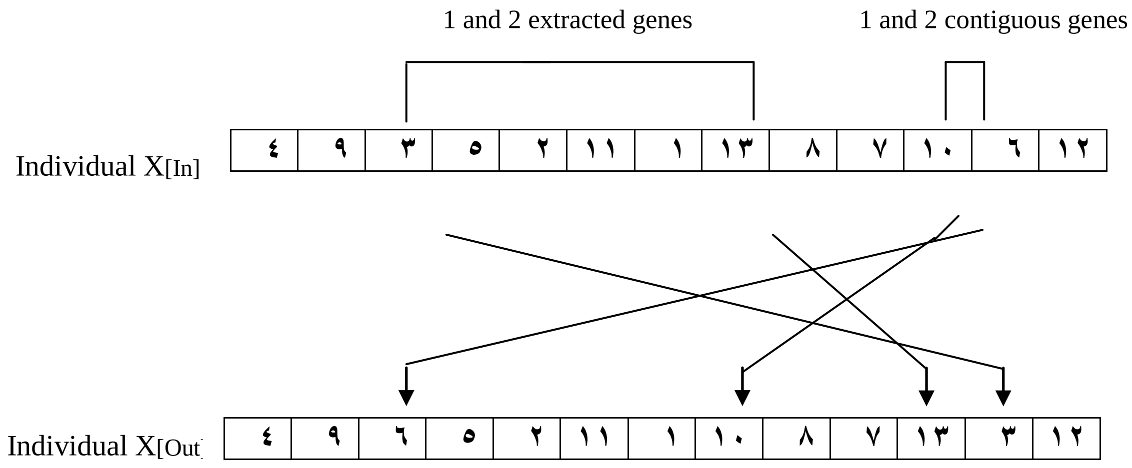


Figure 1-a*: Separation genetic operator*

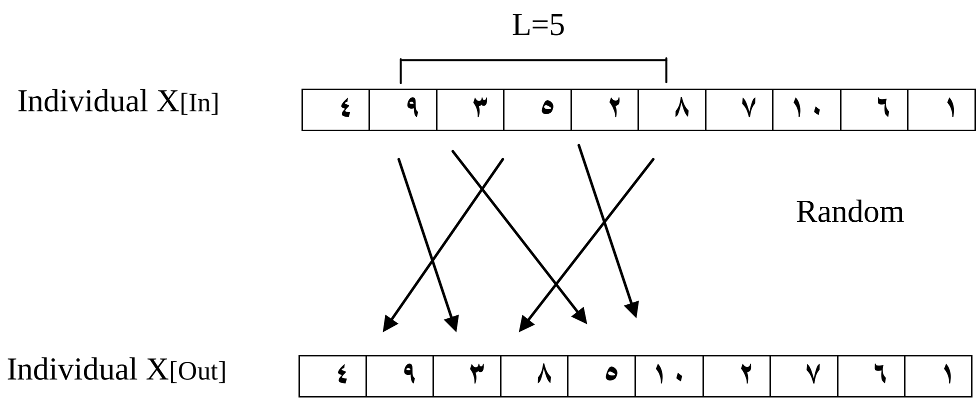


Figure 1-b: ***Annealing genetic operator***

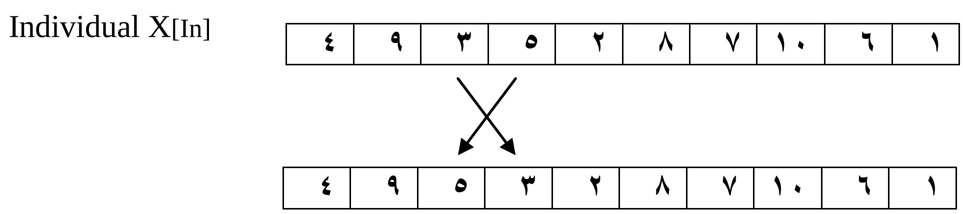


Figure 1-c: ***Swap genetic operator***

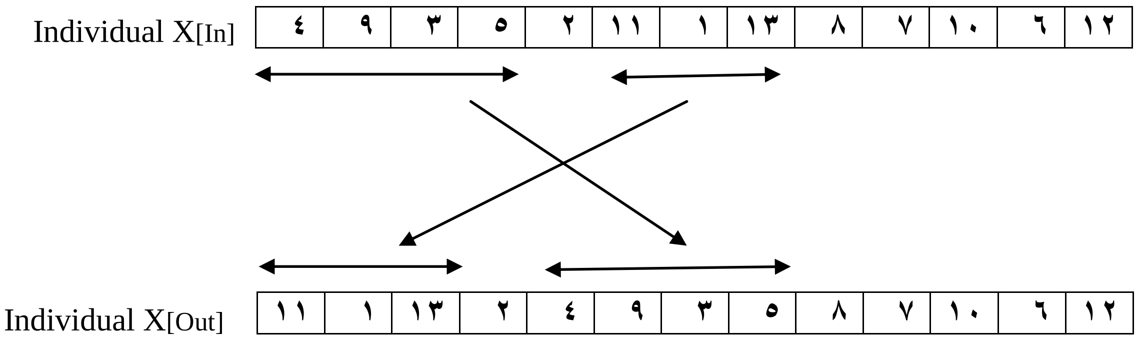


Figure 1-d: ***Block-Swap genetic operator***

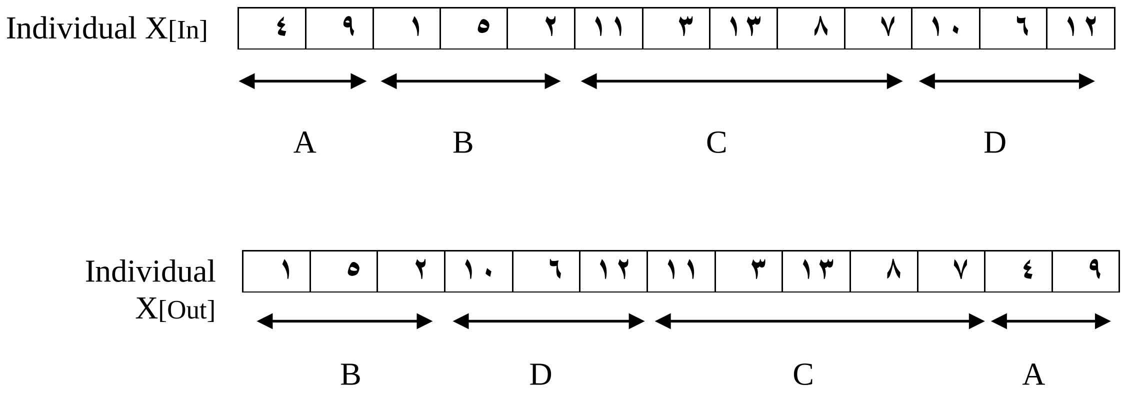


Figure 1-e: ***Shuffle genetic operator***

Figure 1: Suggestion methods

5- The Suggestion Algorithm:

1. Read the face image name
2. Input the population size, generation number
3. Input the number of implementation (10 or 30 or 50)
4. Full the population to the random value of the face image
5. Put the variable gn equal one (gn is the number of generation)
6. Execute the selection and crossover method
7. Execute one method of the new operator of mutation (separation genetic operator or annealing genetic operator or swap genetic operator or Block-Swap genetic operator or Shuffle genetic operator)
8. Replace the old population with new generation
9. Return to step 6 until the variable gn equal to generation number
10. Print the number of implementation of the face image
11. Display the face image
12. end

The general sketch for the suggested algorithm was shown in figure (2).

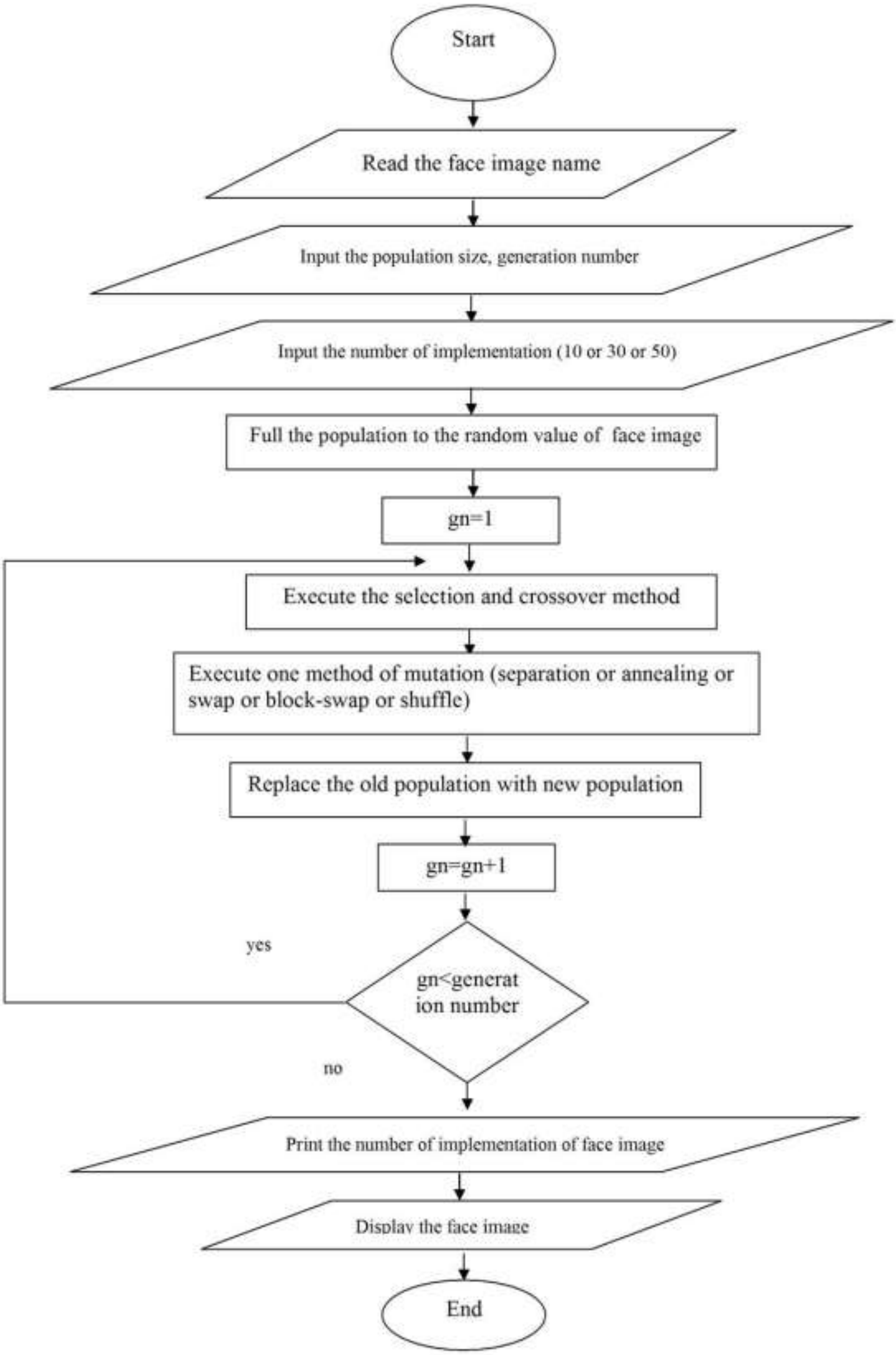


Figure 2: The flow chart of the suggestion algorithm

6- Comparison Between Suggestion Methods:

the application of the five suggested methods was applied on a set of chosen face images. The research dealt with each image alone and the work was divided into three stages: first stage when implementation number is low, that is, implementation number equals 10, Second stage when implementation number is intermediate, that is, implementation number equals 30, third stage when implementation number is high, that is, implementation number equals 50. The results shown in table (1) were gotten.

The chosen face images were displayed before and after applying the suggested algorithm as it is shown in figures (3), (4), (5).

Then drawing the diagram sketches for the results for each image alone as it is shown in figures (6), (7), (8).

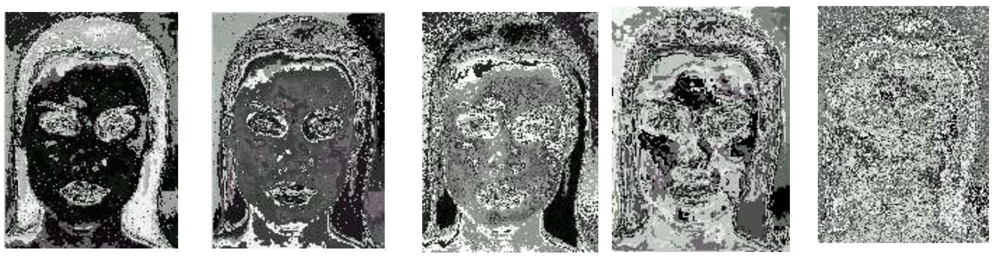
Table (1): Display the results of three face images after execution the suggestion algorithm

Number of Implementation	*Face image Name*	*Separation Genetic operator*	*Annealing Genetic operator*	*Swap Genetic operator*	*Block-Swap Genetic operator*	*Shuffle Genetic operator*	*The Best Methods From Result*
10	Face1	10	10	10	10	10	Equal
30	Face1	30	30	29	28	30	Separation, Annealing, Shuffle
50	Face1	50	49	47	46	49	Separation
10	Face2	10	10	9	8	10	Separation, Annealing, Shuffle
30	Face2	27	26	22	22	30	Shuffle
50	Face2	45	43	37	36	45	Separation, Shuffle
10	Face3	10	10	10	10	10	Equal
30	Face3	30	28	27	28	30	Separation, Shuffle
50	Face3	50	45	43	44	46	Separation

Original Image
Face1

A=10

Separation geneti operator | Annealing genetic operator | Swap genetic operator | Block-Swap genetic operator | Shuffle genetic



B=30

Separation geneti operator | Annealing genetic operator | Swap genetic operator | Block-Swap genetic operator | Shuffle genetic

C=50

Separation genetic operator | Annealing genetic operator | Swap genetic operator | Block-Swap genetic operator | Shuffle genetic operator

Figure 3: Display the original image Face1 and the Face image after execution the suggestion algorithm

The number of Implementation:
A=10, B=30, C=50

Original Image
Face2

Separation genetic operator

Annealing genetic operator

Swap genetic operator

Block-Swap genetic algorithm

Shuffle genetic operator

A=10

Separation genetic operator

Annealing genetic operator

Swap genetic operator

Block-Swap genetic algorithm

Shuffle genetic operator

B=30

Separation genetic operator

Annealing genetic operator

Swap genetic operator

Block-Swap genetic operator

Shuffle genetic operator

C=50

Figure 4: Display the original image Face2 and the Face image after execution the suggestion algorithm

The number of Implementation:
A=10, B=30, C=50

Original Image
Face3

A=10

Separation genetic operator	Annealing genetic operator	Swap genetic operator	Block-Swap genetic operator	Shuffle genetic operator

B=30

Separation genetic operator	Annealing genetic operator	Swap genetic operator	Block-Swap genetic operator	Shuffle genetic operator

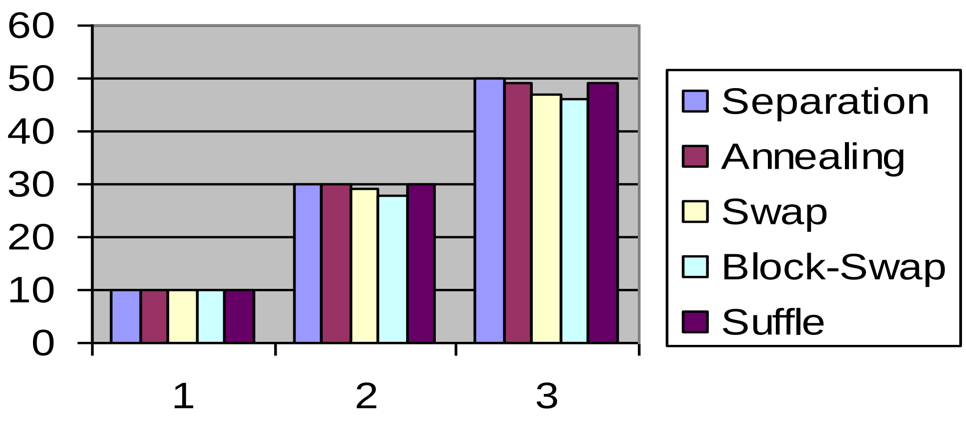
C=50

Separation genetic operator	Annealing genetic operator	Swap genetic operator	Block-Swap genetic operator	Shuffle genetic operator

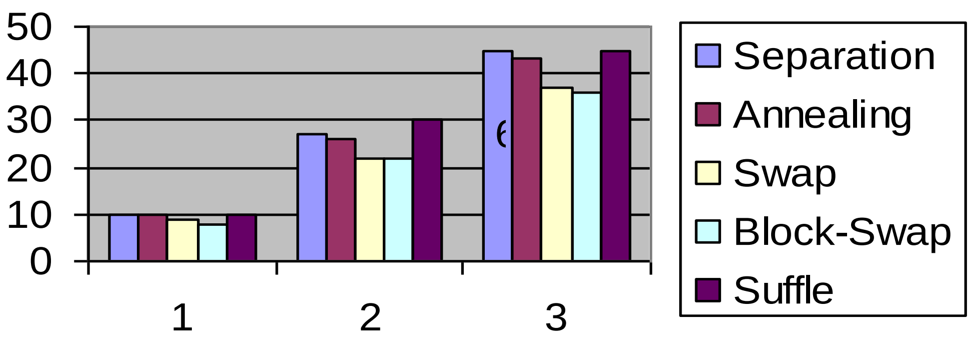
Figure 5: Display the original image Face3 and the Face image after execution the suggestion algorithm

The number of Implementation:
A=10, B=30, C=50

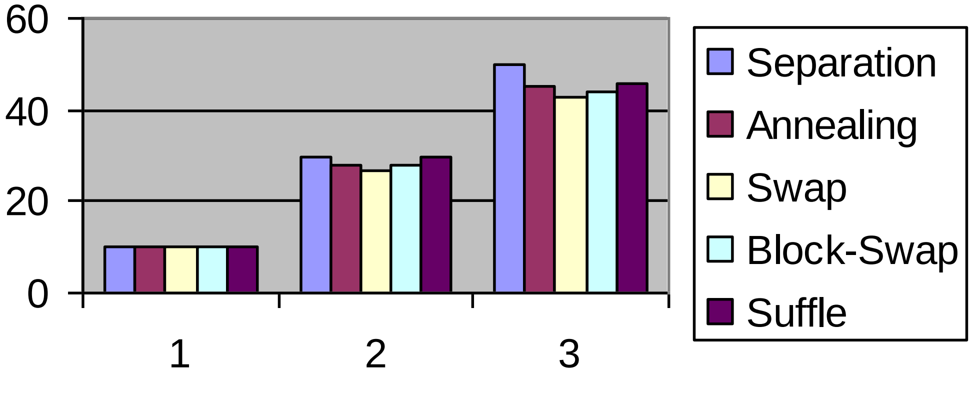
Figure(6): Digram to display the result of Face1 Image



Figure(7): Digram to display the result of Face2 Image



Figure(8): Digram to display the result of Face3 Image



7- Conclusion:

Genetic algorithms (GA) are search algorithms based on the mechanics of natural selection and natural genetics. On the basis of the idea of survival of the fittest, they combine fittest string structures with a structured yet randomized information exchange, to form a search algorithm with some of the innovative flair of human search.

A GA is implicitly parallel and is a randomized algorithm whose results are governed by probabilistic transition rules rather than deterministic rules, a GA operates on several solutions simultaneously, gathering information from current search points and using it to direct subsequent searches which makes a GA less susceptible to the problems of local optima and noise.

The power of a GA lies in its ability to exploit, in a highly efficient manner, information about a large number of individuals.

After applying the suggested methods on the chosen face images (Gray level images) and comparing between them, the following results were given:

1. Separation genetic operator method got the best results.
2. Shuffle genetic operator method got the second cases in results, but the resulted image was deformed because it depends on dividing the chromosome into group of blocks then performing rearrangement which leads to the deformation of the resulted image.
3. Annealing genetic operator method got the third cases in results.
4. Swap genetic operator methods got the fourth cases in results.
5. Block-swap genetic operators method got the least results.
6. The optimal design obtained using combinatorial algorithms such as GA.
7. GA have an advantage over other methods in obtaining multiple solutions of the same quality, thus providing more flexibility to the designer.

8- References:

1-Rennard, Jean. Philippe, “Genetic algorithm viewer: demonstration of a genetic algorithm”, Introduction to genetic algorithms, (2000)

2-Dhar, Vasant and Foster Provost, “Discovering interesting patterns for investment decision making wit a genetic learner overiaid with entropy reduction”, Journal data mining and knowledge discovery, (2000).

3-Huang, Jeffrey and Harry Wechsier, “Eye location using genetic algorithm”, International Conference on audio, (1999).

4-Bala, J. , K. Dejeng and H. Wechsler, “Using learning to facilitate the evolution of features for recognizing visual concepts”, Evolutionary Computation, Vol. 4, No. 3, (1997).

5-Nikravesh, Masoud, F. Aminzade, “Past, present and future intelligent reservoir characterization trends”, Journal of Petroleum Science and Engineering 31, p 67-79, (2001).

6-VESELY, A. , “Neural networks in data mining”, AGRIC. ECON.-CZECH. 49, P 427-431, (2003).

7-Li, Jin and Edward P. K., “Investment decision making using FGP: A case study”, Proceeding, Congress on Evolutionary Computation, Washington DC, USA, (1999).

8-Allen, Franklin Risto karjalainen, “Using genetic algorithms to find technical trading rules”, Journal of financial economics, P 245-271, (1999).

9-Blenkers, Thomas and Peter Zinn, “Application of genetic algorithms to structure elucidation of halogenated alkanes considering the corresponding”, CCACAA 77, P 213-219, (2004).

10-Cto, Cuong, “A parallel genetic algorithm for single class pattern classification and its application for gen expression profiling in streptomyces coelicolor”,BMC genomics, (2007).

11-Kell, Douglas, “Genetic computing: A potential solution to the data mining and predictive modeling challenge today”, (2000).

12-Chiu, Patrick, Andreas Girgeasohn, “A genetic algorithm for video segmentation and summarization”, IEEE, (2000).

13-Mitchell, Tom M., “Genetic algorithms”, Machine learning and data mining, (2004).

14-Sexton, Randall S. and Naheel A. Sikander, “Data mining using a genetic algorithm trained neural network”, International Journal of Intelligent system in accounting, P 201-210, (2001).

15-Prebys, Eric Krvice, “The genetic algorithm in computer science”, MIT under gradate journal of mathematics, P 165-170, (2000).

16-Goh, Kaisong, “Sexual selection for genetic algorithms”, Artificial Intelligence, P 123-152, (2003).

17-Wang, Wei Yen, “Evolutionary learning of bmf fuzzy- neural networks using a reduced form genetic algorithm”, IEEE, Vol. 33, No. 6, (2003).

18-Pakhira, M. K. , “A hybrid genetic algorithm using probabilistic selection”, Vol. 84, (2003).

19-Barton, Valdes J. and Paul R., “Behavior of similarity based neuro-fuzzy networks and evolutionary algorithms in time series model mining”,International ICONIP, (2002).

20-Turk, Matthew, “A random walk through eigenspace”, IEICEE, Vol. ES4-D, No. 12, (2001).

21-Acerra, Frances and scania de schonen, “Modeling face recognition learning in early infant development”, ESANN, P 129-134, (1999).

22-Buscema, Massimo, “Genetic doping algorithm (GenD): theory and applications”, Expert Systems, Vol. 21, No. 2, (2004).
23-Tayal, Manish C. and Yan Fu, “Optimal design of heat exchanges: A Genetic Algorithm framework”, Ind. Eng. Chem. Res. , P 456-467, (1999).
24-Elliott, Lionel Christopier W., “An Informed operator based genetic algorithm for tuning the reaction rate parameters of chemical kinetics mechanisms”, GECCO, P 945-956, (2004).
25-Ochoa, Gabriela Klaus Jaffe, “Assortative mating in genetic algorithms for dynamic problems”, LNCS, P 617-622, (2005).
26-Mitra, Sushmita, Senior Member, “Data mining in soft computing framework: A survey”, IEEE, Vol. 13, No. 1, (2002).
27-Koray, Korkut, “Mining classification rules by using genetic algorithms with non-random initial population and uniform operator”, Yurk J Elec Engin, Vol. 12, No. 1, (2004)
28-Miller, Meghan T. Ronald M., “Feature selection for computer-aided polyp detection using genetic algorithm”, SPIE, Vol. 5031, (2003).

المقارنة بين معاملات جديدة في الخوارزمية الجينية لتمثيل صورة الوجه

شهلة حازم احمد خروفة

قسم العلوم الأساسية ، كلية طب الأسنان ، جامعة الموصل ، الموصل ، العراق

(تاريخ الاستلام: ١١ / ٣ / ٢٠٠٩ ---- **تاريخ القبول:** ٢٥ / ١٠ / ٢٠٠٩)

الملخص

الخوارزمية الجينية هي خوارزمية متكيفة وفعالة وتعتبر طريقة للبحث عامة وتكون مناسبة لإيجاد الحل عندما تكون مساحة البحث واسعة وتستعمل دالة اللياقة وتمكن من الوصول إلى الحل القريب من المثالي باستخدام معاملات الخوارزمية الجينية الأساسية.

وتكون الخوارزمية الجينية قادرة على إنتاج معلومات عن أي موضوع (صورة الوجه مثلاً) عندما تكون ممثلة بطريقة كفوءة ومفيدة.

وفي هذا البحث تم التعامل مع العمليات الأساسية في الخوارزمية الجينية من انتقاء وتقاطع وطفرة وتم التركيز على الطفرة لما لها من أهمية في التأثير على النتائج. إذ تم اقتراح خمسة طرق جديدة لعملية الطفرة لتمثيل صورة الوجه والطرق هي:

(1- Separation genetic operator, 2- Annealing genetic operator, 3- Swap genetic operator, 4- Block-swap genetic operator, 5- Shuffle genetic operator).

وتم المقارنة وإيجاد أفضل طريقة من هذه الطرق، حيث تضمن البحث تقسيم العمل إلى ثلاثة مراحل، في المرحلة الأولى عندما تكون قيمة التمثيل واطئة ، أما المرحلة الثانية عندما تكون قيمة التمثيل متوسطة ، أما المرحلة الثالثة عندما تكون قيمة التمثيل عالية.

تم عمل برنامج بلغة (Matlab (Version 6.5 لتنفيذ العمل وتم التعامل مع كل صورة على حدى وتم عرض الصورة ورسم المخططات البيانية للصورة الناتجة.