A NEW EVOLUTIONARY ALGORITHM FOR OPTIMIZING CONTENT BASED IMAGE INDEXING METHODS

M. Saadatmand-Tarzjan
Tarbiat Modares University
saadatmand@kiaeee.org

H. Abrishami Moghaddam
K.N.Toosi University of Technology
moghadam@eetd.kntu.ac.ir

Abstract: Optimization of content-based image indexing and retrieval (CBIR) algorithms is a complicated and time-consuming task since each time a parameter of the indexing algorithm is changed, all images in the database should be indexed again. In this paper, a novel evolutionary method called evolutionary group algorithm (EGA) is proposed for complicated time-consuming optimization problems such as finding optimal parameters of content-based image indexing algorithms. Contrary to genetic algorithms (GA), the image database is partitioned into several smaller subsets in the new evolutionary algorithm and each subset is used by an updating process as training patterns for each chromosome during evolution. Each EGA chromosome includes an age gene that implies the progress of the updating process, evolutionary genes that participate in evolution and history genes that save previous information of the updating process. Also, a new fitness function is defined to evaluate the fitness of the chromosomes with different ages in each generation. We used EGA to optimize the quantization thresholds of the wavelet correlogram algorithm for CBIR. The optimal quantization thresholds computed by EGA improved significantly all the evaluation measures including average precision, average weighted precision, average recall, average rank, and standard deviation of rank for the wavelet correlogram algorithm.

Key Words: Evolutionary Group Algorithm, Content-Based Image Indexing and Retrieval, Wavelet Correlogram, Global Optimization.

1. Introduction

Storage and retrieval of images in digital image libraries and other multimedia databases become a real demand in industrial, medical, and other applications in recent years. Content-based image indexing and retrieval (CBIR) is considered as a solution. In such systems, in the indexing algorithm, some features are extracted from every picture and stored as an index vector [1]. Then, in the retrieval algorithm, every index is compared (using a similarity criterion) to find some similar pictures to the query image [2].

The aim of enhancing the performance of a CBIR system is usually increasing the similarity between the CBIR and user retrieved images from an image database (imagebase), for all query images. There are generally two ways to enhance the performance of CBIR systems: enhancing the retrieval and indexing algorithms.

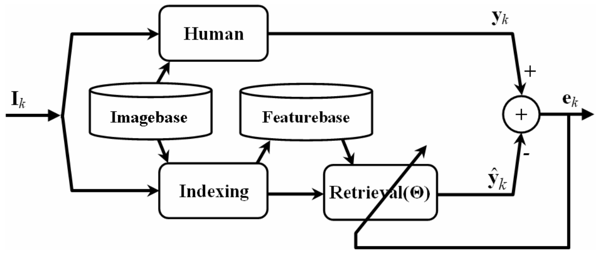


Fig. 1. General block diagram of the retrieval algorithm enhancement process in a CBIR system.

1.1 Retrieval algorithm enhancement

General block diagram of a retrieval algorithm enhancement process is illustrated in Fig. 1. Initially, all images of an imagebase are indexed and their indices are stored in a feature database (featurebase). Then, at each step of the enhancement process, similar images to each query are retrieved using the current retrieval algorithm parameters. Finally, the retrieval algorithm parameters are modified such that the similarity between the CBIR and user retrieved images ($\mathbf{y}_k$ and $\hat{\mathbf{y}}_k$, respectively) for all queries is maximized.

If only one query image ($\mathbf{I}_k$=$\mathbf{I}$) is used in the enhancement process, the retrieval algorithm parameters will be optimized only for that query image. In this case, the block diagram of Fig. 1 corresponds to relevance feedback (RF) approach. RF is a general method to enhance retrieval results using user's feedbacks [3].

1.2 Indexing algorithm enhancement

Here, enhancing the indexing parameters only for one query image is not meaningful (Fig. 2). Accordingly, the aim of enhancement is global optimization of the indexing algorithm parameters.

Optimization of the indexing algorithm is a more difficult task compared to the retrieval algorithm enhancement. Because each time the indexing algorithm parameters are modified, all images of the reference imagebase should be indexed again. Therefore, this optimization process has a large computational cost, particularly for large imagebases. Actually, there is no published work focused on optimization of the indexing algorithm parameters.

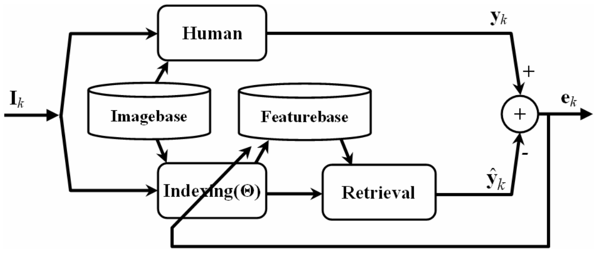


Fig. 2. General block diagram of the indexing algorithm enhancement process in a CBIR system.

A number of researchers used a contextual texture classifier to overcome the above drawback [4]. This classifier categorizes the images into several contextual categories such as picture, painting, portrait, etc. In the indexing algorithm, considering the attachment category of each image, an appropriate index is constructed. Then, in the retrieval algorithm, this index is compared to the indices of the images in same category. Although the above classifier may improve the retrieval algorithm performance, it has several drawbacks such as training cost of the contextual classifier (typically a neural network); and important inaccuracies caused by miss-classifications.

1.3 Optimization Methods

Optimization methods may be divided into two categories: variational approaches such as steepest descent [5]; and global optimizing approaches such as genetic algorithms (GA) [6].

The variational approaches can not be used for parameter optimization of the indexing algorithms; since their cost functions should be convex and differentiable. Among global optimizers, evolutionary approaches such as GA seem to be more adapted to the indexing algorithm optimization problem. However, optimizing the indexing algorithm parameters by typical GA as well as other evolutionary optimizers is a trivial task, because of their huge computational cost. For example, when GA are used, for evaluation of each chromosome, all images of the reference imagebase should be indexed again (see Section 1.2). In other words, if indexing of the reference imagebase takes 20 minutes and GA generates 1200 chromosomes during evolution, its entire computational time will be 400 hours.

1.4 Evolutionary Group Algorithm

In this paper, a novel evolutionary method called evolutionary group algorithm (EGA) for optimization of the indexing algorithm parameters is proposed. Compared to GA, EGA has an important advantage: the evolutionary process in EGA is several times faster than GA.

In EGA, the imagebase is partitioned into several subsets and each subset is used by an updating process as training patterns for each chromosome during evolution. Each EGA chromosome consists of an age gene that implies the progress of the updating process, evolutionary genes that participate in evolution, history genes that save previous states of the updating process, and evaluation genes which save the evaluation function values (simply evaluation values) in different ages. Furthermore, in each generation, a new fitness function evaluates the fitness of the chromosomes with different ages.

We applied EGA for optimizing the quantization thresholds of the wavelet correlogram CBIR algorithm [7]. The optimal quantization thresholds given by EGA improved all the evaluation measures including average precision, average weighted precision, average recall, average rank, and standard deviation of rank for the wavelet correlogram algorithm.

1.5 Paper Outline

The remainder of the paper is organized as follows: Section 2 presents the proposed EGA. In Section 3, EGA is used to optimize quantization thresholds of the wavelet correlogram CBIR algorithm and the simulation results are given. Finally, we conclude in Section 4.

2. EGA Principles

Optimizing the indexing parameters by GA is impractical; since it requires reindexing all images in the reference imagebase for each chromosome which means a very large computational cost. In EGA, the reference imagebase is partitioned into several subsets. In each generation, only one subset is indexed using each chromosome in the population. Consequently in EGA, in contrast to GA, the evolution proceeds during indexing the image subsets of the reference imagebase. This will significantly reduce the computational cost of EGA compared to GA.

2.1 Reference imagebase

In this paper, a subset of COREL database [4] is used as the reference imagebase (**D**). This imagebase consists of 10 image categories ($A = 10$) as listed in Table I, and each category includes $C = 100$ images. Therefore, in each query, the first C retrieved images should be ideally belonged to the similarity category of the query image.

2.2 Reference imagebase partitioning

All images of the reference imagebase are partitioned to L different subsets such that each subset includes $\eta = C/L$ images from each similarity category.

2.3 Proposed Chromosomes

In EGA, the chromosomes are defined as follows:

$$\mathbf{\Theta}'_j = \{g_j, \mathbf{\Theta}_j^{\text{evn}}, \mathbf{\Theta}_j^{\text{his}}, \hat{\mathbf{J}}_j\}, \qquad j = 1,2,\ldots,M \qquad (1)$$

where g_j, $\mathbf{\Theta}_j^{\text{evn}} = [\theta_{j,0}, \theta_{j,1}, \ldots, \theta_{j,N-1}]$, $\mathbf{\Theta}_j^{\text{his}}$, and $\hat{\mathbf{J}}_j$ represent the age gene, evolutionary genes, history genes, and evaluation genes of the *j*-th chromosome, respectively. Evolutionary genes supply inheritance characteristics like the genes of chromosomes in GA. The age is a new gene which enables the chromosome to treat time during evolution. The history and evaluation genes are complementary to the age since they enable the chromosome to treat changes of the environmental conditions as well as time during evolution.

2.4 Updating chromosomes

A chromosome updating process (CUP) is used in EGA in order to update the genes via indexing the new images of the next image subsets. CUP proceeds as follows:

1. *S* mature chromosomes, whose ages are smaller than *L*, are randomly selected from the current population as well as all immature chromosomes to make the set *P* (mature and immature chromosomes will be defined in Section 2.7).
2. The history genes of these chromosomes are extended as follows:

$$\forall j \in \mathbf{P}: \qquad \mathbf{\Theta}_j^{\text{his}} = [\mathbf{\Theta}_j^{\text{his}}, {}^j\mathbf{F}_1^{g_j}, \ldots, {}^j\mathbf{F}_{|\mathbf{D}|_L}^{g_j}] \qquad (2)$$

$${}^j\mathbf{F}_k^i = \text{Indexing}(\mathbf{I}_k^i; \mathbf{\Theta}_j^{\text{evn}}), \;\; i = 0, \ldots, g_j - 1, \;\; k = 1, \ldots, |\mathbf{D}|_L \qquad (3)$$

   where $\mathbf{I}_k^i$ indicates the *k*-th image from the *i*-th subset and $|\mathbf{D}|_L$ is the total number of images in each subset that is equal to $|\mathbf{D}|/L$ where |.| returns the cardinality of a set.
3. The evaluation genes of above chromosomes are also extended as follows:

$$\forall j \in \mathbf{P}: \qquad \hat{\mathbf{J}}_j = [\hat{\mathbf{J}}_j, \hat{J}(\Theta'_j, g_j + 1)] \qquad (4)$$

   where the evaluation function $\hat{J}(\Theta'_j, g_j)$ (that should be minimized) is defined as follows:

$$\hat{J}(\Theta'_j, g_j) = \frac{1}{2C|\mathbf{D}|}\left(\frac{L}{g_j}\right)^2 \sum_{i=0}^{g_j-1} \sum_{k=1}^{|\mathbf{D}|_L} \sum_{p=1}^{g_j\eta} \sum_{q=1}^{g_j\eta} \delta({}^j y_{k,p}^i, {}^j\hat{y}_{k,q}^i) \qquad (5)$$

   where, $\delta(\cdot)$ is the Kronecker delta function, and ${}^j y_{k,p}^i$ is the index of the *p*-th similar image based on Minkowski distance measure [8] of rank *m* in the local featurebase of *j*-th chromosome (history genes $\mathbf{\Theta}_j^{\text{his}}$) for query image $\mathbf{I}_k^i$. Also, ${}^j\hat{y}_{k,p}^i$ is defined the same as ${}^j y_{k,p}^i$ but with ideal distance measure that matches with the user idea about similarity between images.
4. The age genes of these chromosomes are increased by one:

$$\forall j \in \mathbf{P}: \qquad g_j = g_j + 1 \qquad (6)$$

   Therefore, the age gene indicates the progress of the updating process for each chromosome.

2.5 Generation Model

We use the GENITOR model [9] as generation model for EGA. According to this model, only two offspring are produced in each generation and they replace the two chromosomes that have the smallest fitness values in the current population (the fitness function will be introduced in Section 2.6).

2.6 Fitness function

The fitness of each chromosome is a measure to compare it with other chromosomes in a same population and it is used by the selection operator. In simple GA, the fitness function is defined as a function of evaluation values for all chromosomes [6]. Such a simple definition is not useful in EGA, since the chromosomes in a same population have different ages here and are not in same conditions for being comparable. In general, comparing the evaluation values of the chromosomes with various ages is not valid for the following reasons:

1. Chromosomes in different ages have local feature and image-bases with different cardinalities.
2. According to (5), the retrieval results for the chromosomes with various ages have different dimensions.
3. Older chromosomes are more valuable than younger ones since they have been competed more with other chromosomes during evolution.

To overcome the above difficulties, we propose a new fitness computation algorithm (for mathematical discussion, refer to [10, 13]):

1. *k* is set equal to the threshold of puberty *λ* (see Section 2.7).

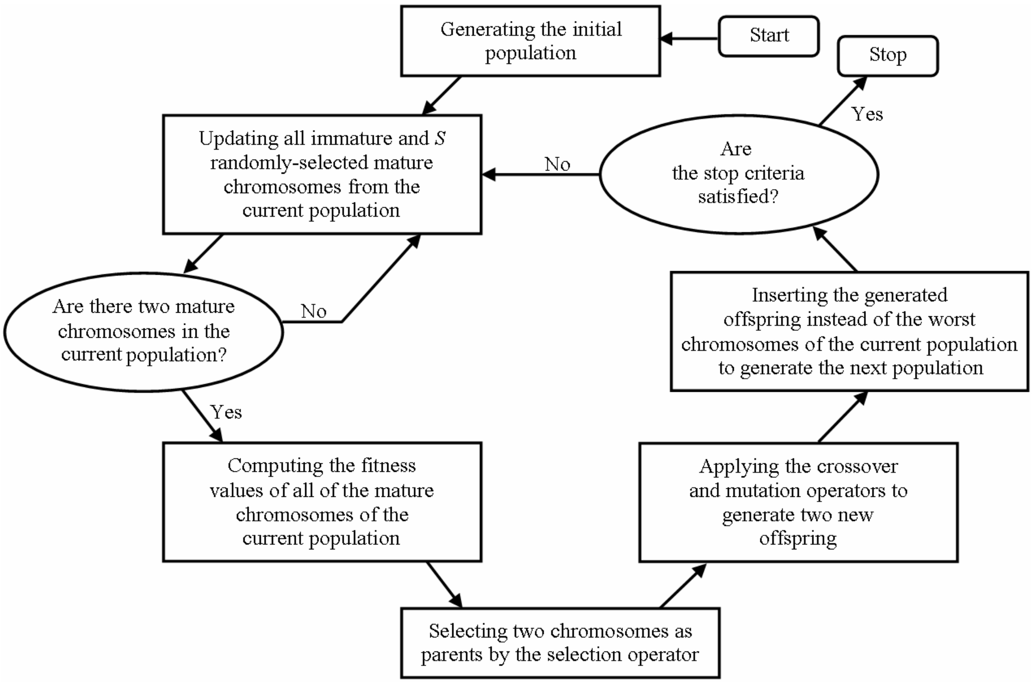

Generating the initial population
Start
Stop
Yes
Updating all immature and S randomly-selected mature chromosomes from the current population
No
Are the stop criteria satisfied?
Are there two mature chromosomes in the current population?
No
Inserting the generated offspring instead of the worst chromosomes of the current population to generate the next population
Yes
Computing the fitness values of all of the mature chromosomes of the current population
Applying the crossover and mutation operators to generate two new offspring
Selecting two chromosomes as parents by the selection operator

Fig. 3. EGA flowchart.

2. The fitness of the immature chromosomes (with the age smaller than λ) is set to zero:

$$\forall k \in \mathbf{Q}_j : \quad P(\Theta'_k) = 0, \qquad j = 0,1,\ldots,\lambda-1 \tag{7}$$

   where $\mathbf{Q}_j$ is a set of all chromosomes with age j.

3. The fitness of all chromosomes that belong to the same-age class $\mathbf{Q}_j$ is computed as follows:

$$P(\Theta'_k; k \in \mathbf{Q}_j) = \left(1 - \sum_{i=1}^{j-1} P(\mathbf{Q}_i)\right) \frac{\hat{J}_{k,j}}{\sum_{i \notin \mathbf{\Gamma}_{j-1}} \hat{J}_{i,j}} \tag{8}$$

4. The total fitness of the same-age class $\mathbf{Q}_j$ is computed by (9):

$$P(\mathbf{Q}_j) = \sum_{k \in \mathbf{Q}_j} P(\Theta'_k, k \in \mathbf{Q}_j) = \sum_{k \in \mathbf{Q}_j} P(\Theta'_k) \tag{9}$$

5. Set j=j+1.
6. Steps 3-5 are repeated until the fitness of all chromosomes is computed.

2.7 Mature and Immature Chromosomes

For each chromosome, the cardinality of the local featurebase (history genes) is related to its age. Therefore, the evaluation value of a chromosome given by (5) will be valid, if its age is larger than the threshold λ. We will call these chromosomes as mature and the chromosomes whose age is smaller than λ as immature. Because of their imprecise evaluations, the immature chromosomes can not be used as parents to generate new offspring during evolution and so, their fitness is always set equal to zero (see Section 2.6). Additionally, in each stage of CUP, all immature chromosomes are kept in the population and are updated until they become mature and generate offspring (see Section 2.4).

2.8 EGA Flowchart

The EGA flowchart is shown in Fig. 3. This evolutionary algorithm initially generates a random population (contains M chromosomes). The chromosomes of the current population are then updated by CUP and the process is repeated until there are at least two mature chromosomes. In the next stage, based on the fitness of chromosomes, two mature chromosomes are selected from the current population as parents by a selection operator. Two offspring are then generated by the parents using crossover and then mutation operators. Finally, a new population is generated by replacing two chromosomes that have the smallest fitness by the new offspring. The above procedure is repeated until a stop criterion is satisfied.

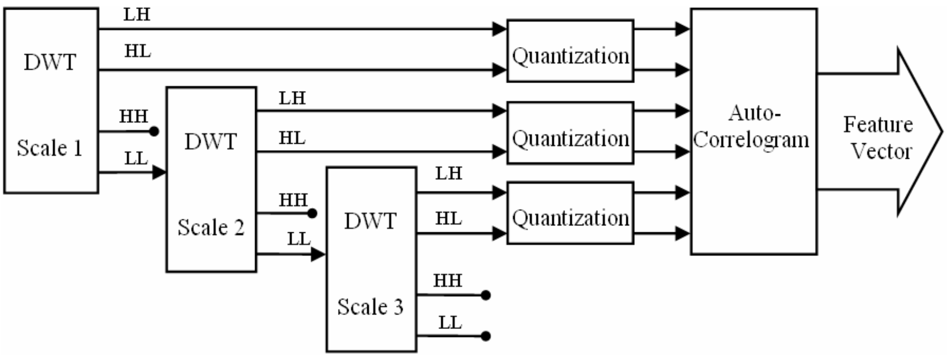


Fig. 4. Block diagram of the wavelet correlogram indexing algorithm.

2.9 EGA and GA Computational Volumes

It can be simply shown that the computational volume of GENITOR-based GA with respect to EGA is given by:

$$\frac{2L}{S+2\lambda}\left(\frac{1+1/\alpha}{1+L/4\alpha}\right) \le \xi_1 = \lim_{k\to\infty}\frac{T[GA_{\mathrm{GENITOR}}(k;M)]}{T[EGA(k;M,S,\lambda)]} \le \frac{2L}{S+2\lambda} \tag{10}$$

For $\alpha \succ\succ L$ we have:

$$\xi_1 \approx \frac{2L}{S+2\lambda} \tag{11}$$

where α is the computational volume of indexing algorithm with respect to evaluation function (for mathematical discussion, refer to [10, 13]).

3. Experimental Results

In this paper, EGA is used to optimize the quantization thresholds of the wavelet correlogram indexing algorithm.

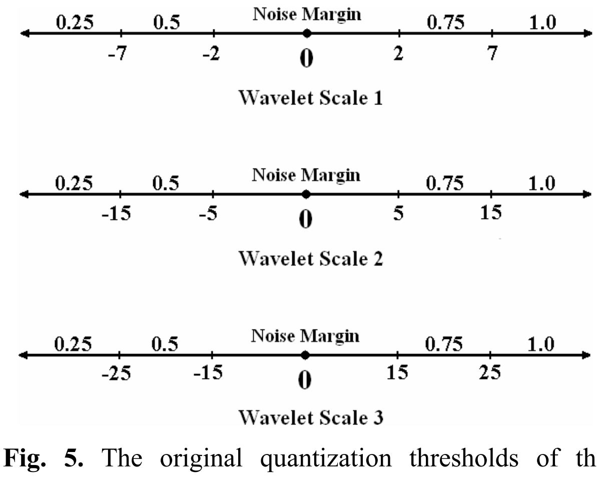


Fig. 5. The original quantization thresholds of the wavelet correlogram algorithm [11].

3.1 Wavelet Correlogram

Wavelet correlogram, recently introduced by Abrishami Moghadam *et al.* [7], is a wavelet-based approach for CBIR. The block diagram of the wavelet correlogram indexing algorithm is shown in Fig. 4. As illustrated, the wavelet coefficients of each image are first computed in three consecutive scales and then quantized using quantization thresholds (Fig. 5). After that, the autocorrelogram of the wavelet coefficients in each scale is computed using only LH and HL matrices. The resultant coefficients are finally used for constructing the image feature vector.

3.2 Optimizing quantization thresholds

To optimize the quantization thresholds of the wavelet correlogram indexing algorithm by EGA, the parameters M, L, λ, S are experimentally chosen as 150, 100, 5, and $0.2\times M=30$, respectively. Also, evolutionary genes of each chromosome are defined as illustrated in Fig. 6.

The initial population is generated based on seeding method [12]. Moreover, we used the tournament selection operator (with the tournament size $M/7$), one-point mathematical crossover operator, and typical mutation operator (with the mutation probability $P_m = 0.01$) [6]. The parameter α was experimentally estimated as α=180. Consequently, using (10), the relative computational volume of GENITOR-based GA with respect to EGA was $4.41 \le \xi_1 \le 5$.

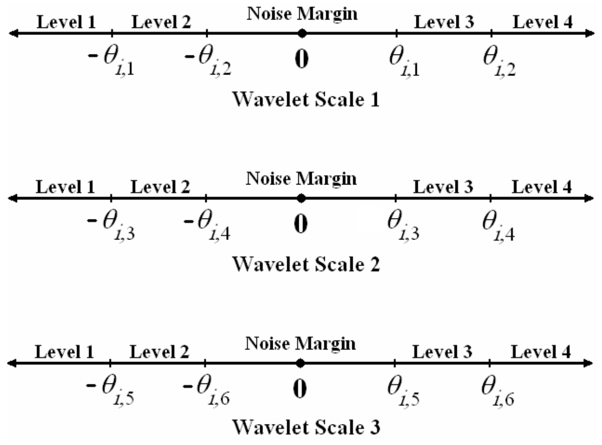

Level 1
Level 2
Noise Margin
Level 3
Level 4
$-\theta_{i,1}$ $-\theta_{i,2}$ 0 $\theta_{i,1}$ $\theta_{i,2}$
Wavelet Scale 1
Level 1
Level 2
Noise Margin
Level 3
Level 4
$-\theta_{i,3}$ $-\theta_{i,4}$ 0 $\theta_{i,3}$ $\theta_{i,4}$
Wavelet Scale 2
Level 1
Level 2
Noise Margin
Level 3
Level 4
$-\theta_{i,5}$ $-\theta_{i,6}$ 0 $\theta_{i,5}$ $\theta_{i,6}$
Wavelet Scale 3

Fig. 6. Quantization thresholds given by the *i*-th chromosome.

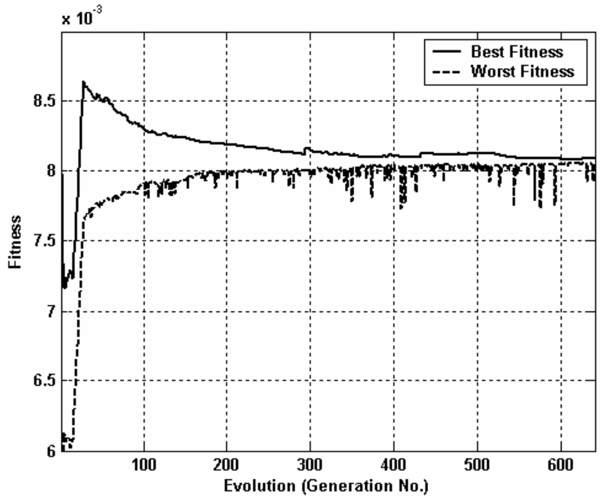


Fig. 7. Fitness variations of the best and worst chromosomes during EGA evolution.

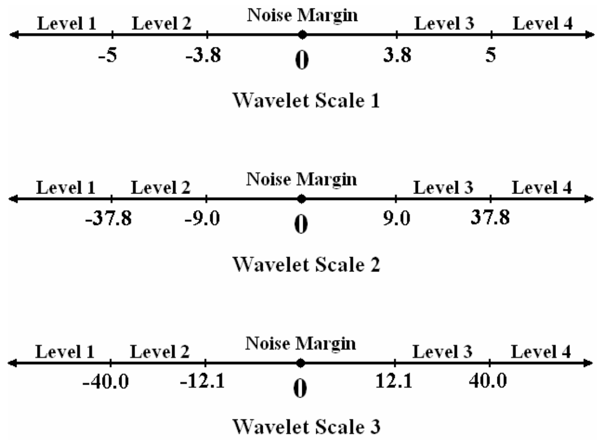


Fig. 8. Optimal quantization thresholds computed by EGA.

3.3 Simulation results

The smooth and reliable variations of fitness probability for the best and worst chromosomes in the population during 640 generations are shown in Fig. 7. The optimal quantization thresholds corresponding to the best (oldest) chromosome computed by EGA are illustrated in Fig. 8.

Tables I demonstrates that the wavelet correlogram performance is considerably improved with EGA-optimized quantization thresholds (Fig. 8) compared to the manually-enhanced ones (Fig. 5) in terms of several evaluation measures [1, 4] including average precision (P_{ave}), average weighted precision ($\overline{P}_{\text{ave}}$), average recall ($R_{\text{ave}}$), average rank ($C_{\text{ave}}$), and standard deviation (STD) of rank (C_{std}). Therefore, EGA was successful to optimize the indexing parameters (the quantization thresholds) of wavelet correlogram in a shorter computational time compared to the conventional evolutionary optimizers such as GA. In Fig. 9, the retrieval results of the wavelet correlogram algorithm with the optimal quantization thresholds (given by EGA) for 2 query images are shown. The precision of the algorithm, for each query image, is appeared at the right-hand side of the graphical user interface.

4. Conclusion

In this paper, a novel evolutionary method called evolutionary group algorithm was proposed to optimize complicated time-consuming problems such as optimizing the parameters of image indexing algorithms. First, the imagebase is partitioned into several subsets in this algorithm. Then, an updating process uses each subset as training patterns for each chromosome during evolution. In EGA, each chromosome includes an age gene indicating the progress of the updating process, evolutionary genes participating in evolution, history genes saving the previous information of the updating process, and evaluation genes storing the evaluation values of the

Table I. Wavelet correlogram performance is significantly enhanced by the EGA-optimized quantization thresholds (indicated by label EGA in the table) compared to the manually-enhanced quantization thresholds (indicated by label WAC in the table).

	Similarity Category	P_{ave} (%)		$\overline{P}_{\text{ave}}$ (%)		R_{ave} (%)		C_{ave}		C_{std}	
		WAC	EGA	WAC	EGA	WAC	EGA	WAC	EGA	WAC	EGA
1	**Africans**	56.7	57.7	68.0	68.2	28.9	31.1	304	282	83	78
2	**Beaches**	49.9	49.3	63.0	61.9	27.9	28.6	345	335	131	131
3	**Buildings**	46.9	50.9	60.1	63.2	29.4	30.5	314	308	142	141
4	**Buses**	83.8	87.1	88.2	91.2	63.4	64.0	110	108	71	79
5	**Dinosaurs**	74.9	74.6	84.5	82.8	25.2	28.8	438	410	69	91
6	**Elephants**	48.5	55.7	63.4	70.7	28.3	30.7	256	235	48	44
7	**Flowers**	84.8	84.3	89.5	88.3	66.0	65.3	116	125	81	82
8	**Horses**	68.9	78.9	81.1	85.9	29.9	39.9	268	264	63	99
9	**Mountains**	39.9	47.2	54.9	60.0	22.1	25.1	345	324	72	79
10	**Food**	48.8	57.1	60.9	67.5	31.0	36.4	249	236	51	57
	Total	60.3	**64.3**	71.4	**74.0**	35.2	**38.0**	275	**263**	129	**127**

chromosome in different ages. Furthermore, EGA defines s new fitness function to evaluate the fitness of the chromosomes with different ages in each generation. EGA has much less computational cost compared to typical GA which enables it to be useful for other time-consuming optimization tasks.

References

[1] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1340-1380, 2000.

[2] V.N. Gudivada and V.V. Raghavan, "Content based image retrieval systems," *IEEE Computer*, 28(9):18-22, 1995.

[3] G.D. Guo, A.K. Jain, and W.Y. Ma, and H.J. Zhang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Trans. Neural Networks*, 13(4):811-820, 2002.

[4] J.Z. Wang, J. Li, and G. Wiederhold, "SIMPLIcity: semantics-sensitive integrated matching for picture libraries," *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(9):947-963, 2001.

[5] O. Nelles, *Nonlinear system identification: from classical approaches to neural networks and fuzzy models*, Berline: Springer-Verlag, 2001.

[6] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, 4(2):65-85, 1994.

[7] H. Abrishami Moghadam, T. Taghizadeh Khajoie, A.H. Rouhi, and M. Saadatmand-Tarzjan, "Wavelet correlogram: a new approach for image indexing and retrieval," *Pattern Recognit.*, 38(12):2506-2518, 2005.

[8] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. San Diego: Academic Press, 2003.

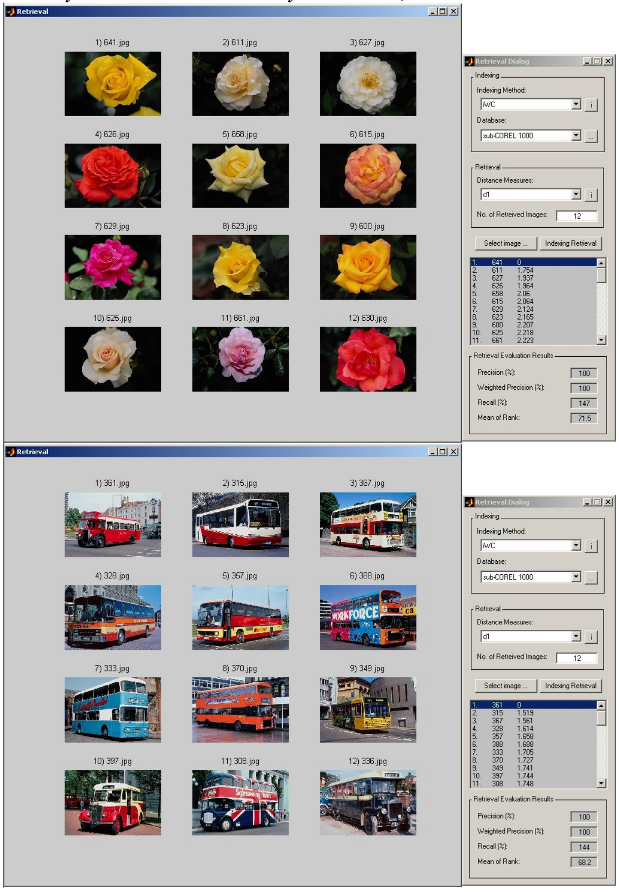
[9] D. Whitley, "The GENITOR algorithm and selective pressure", in *Proc. 3rd Int. Conf. Genetic Algorithms*, Morgan-Kaufmann Ed., pp. 116-121, 1989.

[10] M. Saadatmand, *Improving the wavelet correlogram method for medical image indexing and retrieval*. M.Sc. Thesis, K.N.Toosi University of Technology, Tehran, Iran, 2005.

[11] M. Saadatmand-Tarzjan, H. Abrishami Moghadam, "Improving the wavelet correlogram method for image indexing and retrieval," in *Proc. 6th Conf. Intelligent Systems (CIS'04)*, 2004.

[12] M.A. Lee and H. Takagi, "Embedding apriori knowledge into an integrated fuzzy system design method based on genetic algorithm," in *Proc. 5th IFSA World Congress*, 1993.

[13] M. Saadatmand-Tarzjan, H. Abrishami Moghaddam, "A novel evolutionary approach for optimizing content-based image indexing algorithms," Revised for publication in *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, 2005.



Indexing Method: iWC
Database: sub-COREL 1000
Distance Measures: d1
No. of Retreived Images: 12

```
1.   641   0
2.   611   1.754
3.   627   1.937
4.   626   1.964
5.   658   2.06
6.   615   2.064
7.   629   2.124
8.   623   2.165
9.   600   2.207
10.  625   2.218
11.  661   2.223
```

Precision (%): 100
Weighted Precision (%): 100
Recall (%): 147
Mean of Rank: 71.5

Indexing Method: iWC
Database: sub-COREL 1000
Distance Measures: d1
No. of Retreived Images: 12

```
1.   361   0
2.   315   1.519
3.   367   1.561
4.   328   1.614
5.   357   1.658
6.   388   1.688
7.   333   1.705
8.   370   1.727
9.   349   1.741
10.  397   1.744
11.  308   1.748
```

Precision (%): 100
Weighted Precision (%): 100
Recall (%): 144
Mean of Rank: 68.2

Fig. 9. Retrieval results of wavelet correlogram with EGA-optimized quantization thresholds for 2 different query images.