International Journal of Computer Applications (0975 – 8887)
Volume 46– No.2, May 2012

Comparison of Hybrid and Classical Metaheuristic for Automatic Image Enhancement

Anupriya
Indira Gandhi Institute Of Technology
Delhi, INDIA

Akashtayal
Indira Gandhi Institute Of Technology
Delhi, INDIA

ABSTRACT
Hybrid metaheuristic, an advancement over classical metaheuristic, provides a more effective search methodology. It combines several metaheuristic algorithms into one optimization mechanism. In this paper image enhancement is considered as an optimization problem. Hybrid metaheuristic techniques are used to find the optimum value for a set of parameters of a transformation function, with an aim towards maximizing a fitness function. Three hybrid metaheuristic approaches are employed to find the optimum solution. Results of all three algorithms are compared amongst themselves. Comparison is also shown with classical metaheuristic algorithms and traditional enhancement approach of histogram equalization.

General Terms
Image Processing, Hybrid Metaheuristic

Keywords
Differential Evolution, Genetic Algorithm, Hybrid Metaheuristic, Image Enhancement, Particle Swarm Optimization, Simulated Annealing

1. INTRODUCTION
Image enhancement is a crucial primary step in virtually all image processing applications like image segmentation, image reconstruction, object detection, object classification and image analysis. There is a need to highlight details, improve contrast of images and enhance their visual perception in application areas such as biomedical image analysis, industrial inspection, and criminology and computer vision. Image enhancement methods can be broadly classified into four types of operations: point operations (e.g. contrast stretching, histogram equalization), spatial operations (e.g. median filtering, noise smoothing), transform operations (e.g. homomorphic filtering) and pseudo colouring [1]. Most of the traditional enhancement techniques have strong dependence on the type of image being processed and require subjective human evaluation (of the transformed image). Automatic image enhancement requires an objective evaluation criterion which is applicable to a wide range of image types. In this paper we consider image enhancement as an automatic parameterized process. A parameterized transformation function is used to enhance the image, which is then evaluated by an objective fitness criterion. The role of optimization technique employed is to maximize the objective evaluation function by finding the optimum configuration of parameters of the transformation function. Automatic image enhancement is a complex optimization problem with a large problem instance, requiring the use of metaheuristic techniques. In the past several metaheuristic algorithms have been used for this purpose. A method to enhance contrast of images using genetic algorithm was proposed by Saitoh in [2]. [3] uses genetic algorithm to design a filter for image enhancement. Particle swarm optimization has been used for gray level image enhancement in [4]. Differential evolution based adaptive image enhancement scheme is proposed in [5]. In [6] authors have proposed a method for image enhancement based on genetic algorithm using a subjective evaluation criterion. An image restoration algorithm using simulated annealing has been put forth in [7].

In this paper we present a comparison of the performance of hybrid metaheuristic and classical metaheuristic in the field of image enhancement. Hybrid metaheuristic is a term coined to represent a class of algorithms which use a combination of metaheuristic with some other optimization technique such as Dynamic Programming, Integer Linear Programming (ILP) or other metaheuristic algorithms [8]. Each optimization technique has its own set of capabilities and limitations. The aim of hybrid algorithm is to adopt the advantages of constituent search methodologies, while attempting to eliminate their limitations. In literature, many hybrid metaheuristic approaches have been proposed. [9] proposes to combine Ant Colony Optimization (ACO) with Genetic Algorithm (GA) to improve the search efficiency. A combination of GA and Particle Swarm Optimization (PSO) has been used in [10] in order to achieve lower error at the cost of higher computation time. A literature survey on combination of local search techniques with Constraint Programming (CP) is given in [11].

The hybrid approaches compared in this paper are an integration of population based metaheuristic algorithms and trajectory methods. Population based methods are good at exhaustive exploration of the search space [12]. On the other hand trajectory or single solution based methods have the capability of intensifying search in the promising areas. A hybrid approach combining the two balances the intensification and diversification components to yield a better optimization scheme. We have used a combination of population based methods [viz. Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE)] with single solution based method [viz. Simulated Annealing (SA)]. The comparison between the three hybrid algorithms is made on the basis of objective evaluation of the resultant image. It is also shown that hybrid approach performs better than individual metaheuristic algorithms. The result is also compared with classical enhancement technique of histogram equalization.

The remainder of the paper is organized as follows: Section 2 describes the transformation function and fitness criteria used. Description of classical metaheuristic algorithms is given in Section 3.Section 4 gives details and specifications of hybrid algorithms used in this paper. In Section 5 experimental setups and results obtained are discussed. The paper ends with a conclusion in Section 6.

2. TRANSFORMATION FUNCTION AND EVALUATION CRITERIA

For the purpose of enhancement, a transformation operation is performed on the input image. The value of each pixel in input image is transformed to a new gray scale value in the output image. The quality of transformed image is evaluated using an objective fitness criterion. The fitness criterion is a measure of the contrast of image and the extent to which details (e.g. edges) are enhanced.

2.1 Transformation Function

We use spatial operation for image enhancement. The transformation operation can be represented as

$$v(m, n) = \mathbf{T}[u(m, n)] \quad (1)$$

Where, **T** represents the transformation function,
u (m, n) and v (m, n) are the intensity values of (m, n) th pixel in the input and output image respectively.
The transformation function used in this paper considers both local as well as global statistics for image enhancement. Local statistics help preserve the details which are localized to a small region or a small number of pixels, which might not have been adequately represented in the global statistics.
The transformation function employed [13] is given in equation (2). It is a generalized version of statistical scaling technique [1].

$$v(m,n) = T[u(m,n)]$$

$$= \frac{k*M}{(\sigma(m,n)+b)}[u(m,n) - c*\mu(m,n)] + \mu(m,n)^a \quad (2)$$

Where, a, b, c, k are positive real constants,
M is the global mean of the image given by equation (3)

$$M = \frac{1}{H*V}\sum_{x=0}^{H-1}\sum_{y=0}^{V-1} u(x,y) \quad (3)$$

H and V are the horizontal and vertical size of the image respectively.
$\mu(m,n)$is the local mean of the pixel values contained in a window of size N*N centred around the pixel (m, n) given by equation (4)

$$\mu(m,n) = \frac{1}{N*N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} u(x,y) \quad (4)$$

$\sigma(m, n)$is the local standard deviation in a window of size N*N centred around the pixel (m, n) given by equation (5)

$$\sigma(m,n) = [\frac{1}{N*N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[u(x,y) - \mu(m,n)]^2 \ ^{0.5} \quad (5)$$

In equation (2), use of constant c ensures that only a fraction of the local mean is subtracted from original pixel value. The last additive term has a brightening effect on image due to the term's dependence on local mean. A positive real value of constant b ensures a non-zero denominator in case of zero standard deviation.
The purpose of optimization algorithm is to find the optimum configuration of parameters a, b, c, k, suited to the image under consideration.

2.2 Objective Evaluation Function

Automatic image enhancement needs an objective fitness criterion which can aid in judgment of image quality without human intervention. We use a fitness criteria proposed in [13]. It is given in equation (6) below

$$F(par) = \ln[\ln[E(Is(par)) + e]] * \frac{n_{ep}(I(par))}{H*V} * e^{H(I(par))} \quad (6)$$

Where, *par* represents a set of values of four constants a, b, c and k.*I (par)*is the transformed image generated using parameter set *par*. *Is (par)* is the image obtained by applying sobel (edge) operator to transformed image. *e*is the Euler constant and *F* is the value of objective function.

The evaluation function takes into account the following measures of image quality.

- *H(*):* Modified version of entropy of Image

$$H = -\sum_{n=1}^{G} fn \log(fn) \quad (7)$$

Where, G is the number of gray scale levels in the image,
fn is the frequency of pixel having the intensity n,
Higher the entropy value, more evenly are the gray levels distributed in an image leading to a better appearance. Higher entropy value means that the image does not have extreme contrast ensuring a natural look.

- n_{ep} : Number of edge pixels detected using sobel operator and automatic thresholding

- *E (*):* Intensity of edge pixels given by equation (8) obtained by applying sobel operator to transformed image

$$E(I) = \sum_m \sum_n [\delta h(m,n) + \delta g(m,n)]^{0.5} \quad (8)$$

Where,

$$\delta g(m,n) = I(m-1,n+1) + 2I(m,n+1) + I(m+1,n+1) - I(m-1,n-1) - 2I(m,n-1) - I(m+1,n-1)$$

$$\delta h(m,n) = I(m+1,n+1) + 2I(m+1,n) + I(m+1,n-1) - I(m-1,n+1) - 2I(m-1,n) - I(m-1,n-1)$$

Greater is the number of pixels and higher is their intensity, better are the details represented in the transformed image. Double logarithm operator is used to prevent the value of edge intensity from dominating the fitness expression.

3. CLASSICAL METAHEURISTIC

A brief description of traditional metaheuristic algorithms used in our hybrid approach is given in this section.

3.1 Genetic Algorithm (GA)

GA is an evolutionary algorithm proposed in 1975 by John Holland [14]. It derives inspiration from the biological phenomena of evolution and survival of the fittest. GA starts with a randomly selected population of individuals (chromosome). The fitness of each individual is computed using an objective function. From the current population, high fitness individuals are selected for mating using some selection criteria such as roulette wheel, elitism etc. [12]. The selected parents undergo crossover to produce offspring. Crossover combines parents (selected individual) in such a way that each offspring inherits properties from both parents. Examples of crossover operator are uniform crossover, binary crossover and arithmetic crossover. Next, some amount of diversity is introduced in the population via mutation which arbitrarily changes some traits in randomly selected individuals. Next generation is chosen from the parent and offspring population depending on the fitness of individuals. This process iterates till the termination condition (maximum number of generations, minimum fitness etc.) is met. The pseudocode for GA is given in figure 1.

3.2 Particle Swarm Optimization (PSO)

PSO was developed by Kennedy and Eberhart in 1995 [15]. It is based on the swarm (flocking) behaviour of birds. It begins with a randomly selected swarm of particles (equivalent to birds in a flock). Each particle has associated with it a position and velocity vector. Position vector defines the current position of the particle, while the velocity vector determines how the particle moves about in the search space. Each particle updates its position based on the best position acquired by it (pBest) and the best position attained by any particle in the swarm (gBest). This process follows the equations mentioned below.

$$v(t+1) = w(t) * v(t) + c1 * r1\ (gbest - x(t)) + c2 * r2\ (pbest - x(t)) \quad (9)$$

$$x(t+1) = x(t) + v(t+1) \quad (10)$$

Where, v (t) is the velocity of particle at iteration t,
x (t) is the position of particle at iteration t,
c1 and c2 are the acceleration constants that govern the way each particle follow the best particle,
w (t) is the inertial weight at iteration t; iteration starts with a high value of w to search a large space, and it reduces as iteration proceeds, thus leading to intensification,
r1, r2 are random numbers picked up from a uniform distribution.
This iterative process continues till the termination criterion is met. The pseudocode for PSO is given in Figure 2.

3.3 Differential Evolution (DE)

DE is an evolutionary algorithm put forth in 1995 by Storm and Price [16]. It follows a principle similar to GA but uses a different type of crossover operator. Also the way new population is generated/selected is different from GA. Similarto GA, DE also begins with a random initial population. The offspring are generated by applying mutation and crossover operator to individuals of current population. Crossover and mutation in DE is a type of linear combination traditionally using three parents. The process of mutation is illustrated in figure 3. The crossover procedure is shown in figure 4. It uses a crossover constant whose value usually ranges from 0.2-1. It also uses a scaling factor F, F ∈ [0, 1]. From amongst the newly generated offspring population and current population, new generation individuals are selected using their fitness as selection criteria. The pseudocode for DE is shown in figure 5.

```
Initialize population by randomly generating individuals
Repeat
   Evaluate fitness of individuals
   Select parents based on their fitness
   Produce offspring using crossover operator
   Apply mutation operator to diversify population
   Evaluate fitness of offspring
   Select new generation from union of current
population and offspring produced
Until termination criteria met
```

Figure 1. Pseudocode for GA

```
Initialize swarm of randomly generated particles
Initialize velocity vector
Repeat
   Evaluate fitness of each particle
   Select particle with best fitness, gBest
   Select best position of each particle, pBest
   Update velocity vector using equation (9)
   Update position of each particle using equation (10)
Until termination criteria is met
```

Figure 2. Pseudocode for PSO

```
For each individual of population, p (i)
off (i) ← p (j) + F * (p(k) - p(l))
end for
```

Figure 3.Mutation in DE

```
For each gene (j) of chromosome (i),
  If rand < crossover constant
new (i)(j) ← off (i)(j)
else
new (i)(j) ← p (i)(j)
end if
end for
```

Figure 4.Crossover in DE

```
Initialize population by randomly generating individuals
Repeat
   Evaluate fitness of individuals
Perform mutation and crossover
Evaluate fitness of new offspring
   Select new generation from amongst new and current
population based on fitness
Until termination criteria met
```

Figure 5.Pseudocode for DE

```
For current solution s and candidate solution s'
   If fitness (s) > fitness (s')
s← s'
else if exp [(fitness(s') – fitness (s))/kT] < rand
s← s'
end if
end for
```

Figure 6.Selection criteria in SA

Select initial random solution and starting temperature
Repeat
For a fixed number of iterations
Evaluate fitness of solution
Generate candidate solution in neighbourhood of current solution
replace current solution by candidate if criteria in figure 6 satisfied
end for
Reduce temperature based on cooling schedule
Until termination criteria met

Figure 7.Pseudocode for SA

3.4 Simulated Annealing (SA)

SA algorithm was first applied to optimization problem by S. Kirkpatrick et al. [17]. It is a single solution based local search procedure. SA has the advantage of being capable of moving out of local optima. This is possible because unlike above mentioned algorithms,SAaccepts solutions that are not necessarily better than the existing ones. SA derives its inspiration from the annealing process in mechanics. It involves heating a substance to a high temperature followed by gradual cooling until it crystallizes. SA starts with a single initial solution. The neighbourhood of solution is searched to find other candidate solutions. If the fitness of candidate solution satisfies any of criteria specified in figure 6, then it replaces the current solution. The controlling parameter of the algorithm is temperature, T. Initially a high value of T is selected enabling acceptance of large number of candidate solutions. As process continues, temperature is reduced (using an appropriate cooling schedule [18]). This leads to convergence to an optimum solution. The iteration proceeds till a termination criterion such as minimum temperature is reached. The pseudocode for SA is given in figure 7.

4. HYBRID METAHEURISTIC

Population based methods are diversification intensive i.e. they are capable of exploring a large solution space. However they can get stuck in local optima at times. This drawback can be overcome by use of simulated annealing local search technique. Because SA can accept worse solutions also, integrating SA with population based methods can overcome the latter’s limitation.

4.1 Basic Methodology

In our paper we combine population based metaheuristic (GA, DE and PSO) with SA for generating a hybrid algorithm. We start with a high temperature, T (control parameter) appropriate for our image enhancement problem. At each value of control parameter T, an iteration of any of the population based methods is carried out. The new population hence generated is fed into the local search methodology of simulated annealing. Based on the current value of control parameter T and fitness of current and candidate solution, solutions of Evolutionary Computation (EC) are replaced by a neighbourhood solution generated using SA algorithm. After each step, temperature is reduced following the cooling schedule. The process continues till a culmination criterion of minimum temperature is reached.
We use a cooling schedule illustrated in equation (11)

$$T\,(t+1) = \alpha * T(t) \tag{11}$$

Where α ∈ [0, 1], a high value of α is chosen to ensure slow cooling.

Hybrid methodology ensures that we are able to move out of local optima and converge to global optima. The computation time required is higher (because of increased processing) than classical metaheuristic but results in a higher fitness value.

4.2 Template of Hybrid Algorithm

Select algorithm’s parameters
Bounds of solution space, h_b, l_b
Initial temperature T0 ≥ 0
Cooling schedule, cool (t)
Maximum iteration at fixed temperature, N
Objective function f (*)
Population size, NP

Begin
Initialize control parameter T ← T0
Generate initial population, pop
Calculate fitness value for initial population
fitness_current← f (pop)

Repeat
Generate new solution set for concerned population based method
(Refer Section III for procedure)

for i = 1: N
Generate candidate solution, C(s) in neighbourhood of each solution in pop
If selection criteria satisfied
(Refer figure 6)
pop← C(s)
end if
end for

Reduce temperature as per cooling schedule
T ← cool (T)

***Until** termination criteria satisfied*

4.3 Hybrid GA-SA Specifications

GA suffers from the problem of premature convergence. This can be avoided by use of SA algorithm. We have used Real Coded Genetic Algorithm (RCGA). In the genetic algorithm section of hybrid algorithm we use a combination of elitism and binary tournament selection [12]. Elitism (best 6 solution retained) ensures the best individual (solution) searched are not lost while tournament selection exhibits a high selection pressure. Arithmetic crossover [19] used to produce offspring is illustrated in equation (12). This crossover guarantees high similarity between parent and offspring.

$$\begin{aligned} Offspring\,(1) &= rand * parent1 + (1 - rand) * parent2 \\ Offspring\,(2) &= rand * parent2 + (1 - rand) * parent1 \end{aligned} \tag{12}$$

The four parameter values to be optimized form the genes of the chromosome (solution). Each gene represents one of the four parameters.

4.4 Hybrid PSO-SA Specifications

PSO requires a large swarm size for better optimization. The need for higher number of particles can be eliminated by using a local search technique (SA) with PSO. Each particle’s position is represented by a vector in 4-dimnesional search space (each dimension corresponding to one of the four parameters to be optimized). It has associated with it a velocity vector. The inertial weight *w (t)* is varied in accordance with equation (13) given below.

$$w(t+1) = w(t) - [(w_{max} - w_{min})/\max iteration] \quad (13)$$

Inertial weight is varied between 0.2 and 0.6. The maximum change in velocity in one step is limited to 20.The value of both the acceleration constant is kept at a moderate 1.3. Low values of acceleration constants allow particles to roam around randomly, away from target region, while high values result in abrupt movement toward the target regions.

.

4.5 Hybrid DE-SA Specifications

Integrating SA algorithm with traditional DE algorithm builds a superior algorithm. It helps in attaining a better solution by avoiding early convergence. For the DE portion of hybrid algorithm each individual is a vector of four real numbers. We use binomial crossover illustrated in section III. The scaling factor F is set equal to 0.8. The value of crossover constant is chosen to be 0.2.

5. EXPERIMENT AND RESULTS

The results of image enhancement using hybrid algorithms (PSO-SA, GA-SA and DE-SA) are presentedfor four images. They are compared with results obtained using classical (GA, DE and PSO) algorithm as well as with the output of Histogram Equalization (HE). The values shown in table 1 And table 2 are for an average of ten independent runs of each algorithm.

The range of parameters to be optimized i.e. a, b, c, k is as follows:a ∈ [0, 2], b ∈ [0, 1], c ∈ [0, 0.5], k ∈ [0, 2]

The fitness (objective function) value obtained for each of the classical metaheuristic algorithms and histogram equalization is given in Table 1.

For GA and DE a population size of 60 is used. Maximum number of generations for both is 50. For PSO, the swarm size is 40 particles and maximum number of iterations is 50.

Table 1. Fitness Value for Classical Metaheuristic

Image	PSO	GA	DE	HE
Lena	270.45	238.89	217.74	154.62
Cameraman	260.05	255.17	209.74	89.26
Living Room	225.69	190.05	194.44	96.03
Mandril	224.28	146.63	172.41	72.47

The fitness values in Table 1 illustrates that image enhancement produced using metaheuristic technique gives superior results as compared to histogram equalization method. Also the fitness score obtained using PSO technique is better than those obtained using GA or DE (in general) even with a smaller swarm size.

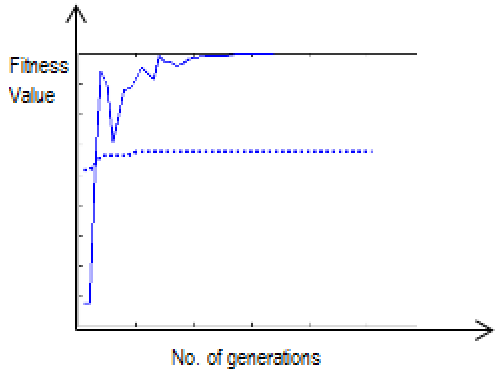

Fig. 8a

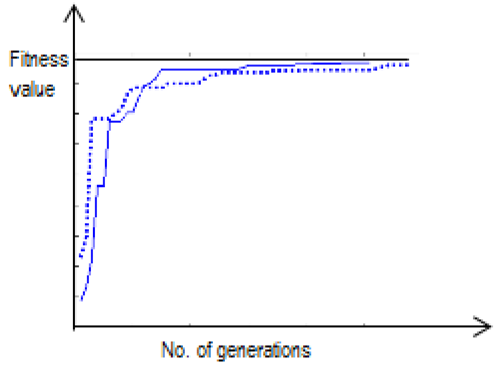

Fig. 8b

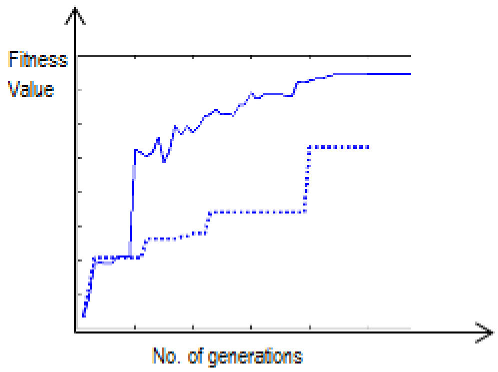

Fig. 8c

Figure 8. Convergence behaviour of hybrid and classical Metaheuristic

……. : Classical Metaheuristic

------- : Hybrid Metaheuristic

a. GA and GA-SA **b. PSO and PSO-SA**

c. DE and DE-SA

The fitness (objective function) value obtained for each of the hybrid algorithms is given in Table 2. For PSO-SA, the swarm size is 15. Population size for DE-SA and GA-SA is taken to be 30.

Table 2. Fitness Value for Hybrid Metaheuristic

Image	PSO-SA	GA-SA	DE-SA
Lena	274.99	280.24	272.37
Cameraman	262.98	263.4	225.6
Living Room	226.76	226.74	227.15
Mandril	228.47	232.63	211.48

The comparison of data in table 1 and 2 shows that the hybrid approach gives better results than standalone metaheuristic algorithms, even with a smaller population size. Also, the improvement is more pronounced for genetic algorithm and differential evolution than for particle swarm optimization.

A comparison of convergence behaviour of hybrid metaheuristic with classical metaheuristic is shown via graphs in figure 8. The graphs are plotted for the value of objective function versus the number of iterations for the image "Lena". As is evident from the figure, GA and DE show a very fast (premature) convergence, whereas their hybridized version with SA gives far better results. The improvement, because of combination with SA, in convergence behaviour of PSO is marginal. Amongst the hybrid algorithms, swarm size required for PSO-SA to give results comparable to GA-SA is lower. Therefore the computation time required for PSO-SA is less than that for DE-SA and GA-SA.

6. CONCLUSION AND FUTURE WORK

In this paper we have compared the performance of hybrid metaheuristic and classical metaheuristic algorithms in the field of image enhancement. The results show that hybrid approach in general produce superior results as compared to classical metaheuristic techniques. Also the result of both hybrid and classical metaheuristic is better than that obtained from traditional method of histogram equalization. Metaheuristic techniques adopt a more generalized approach for image enhancement as the parameters are tuned for each image type individually, unlike histogram equalization which applies same computation scheme to all images. The reason for hybrid approach performing better is that they have the capability of balancing the explorative and exhaustive component of search methodologies. Also, hybrid approach requires a smaller population or swarm size as compared to standalone evolutionary computation techniques. It is seen that the improvement in performance, due to hybridization, is higher in DE and GA than in PSO. This stems from the fact that the problem of premature convergence is more pronounced in DE and GA than PSO. Use of SA helps compensate this drawback of DE and GA.

The optimization used in this paper is based on hybridization approach involving two metaheuristic techniques. We can extend this approach to accommodate more than two search methodologies. Also the optimization can be carried out by use of hyper heuristic. Hyper heuristic aims to search for solution in the metaheuristic space rather than in solution space i.e. it involves finding the best metaheuristic technique that can be employed to a given optimization problem.

7. REFERENCES

[1] A. K. Jain, Fundamentals of Digital Image Processing. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[2] F. Saitoh, Image contrast enhancement using Genetic Algorithm, in Proc. IEEE SMC, Tokyo, Japan, 1999, pp. 899–904.

[3] Jin-Hyuk Hong, Sung-Bae Cho, Ung-Keun Cho, A Novel Evolutionary Approach to Image Enhancement Filter Design: Method and Applications, IEEE Transaction on System, Man and Cybernetics-Part B: Cybernetics, Vol. 39, No.6, Dec 2009

[4] ApurbaGorai, AshishGhosh, Gray-level Image Enhancement By Particle Swarm Optimization, World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)

[5] Qingyun Yang, An Adaptive Image Contrast Enhancement based on Differential Evolution, 3rd International Congress on Image and Signal Processing , (CISP2010)

[6] C. Munteanu, V. Lazarescu, Evolutionary contrast stretching and detail enhancement of satellite images, In Proc. Mendel, Berno, Czech Rep., pp. 94-99, 1999.

[7] Yen-Wei Chen, TatsuroEnokura, and ZenshoNakao, A Fast Image Restoration Algorithm Based on Simulated Annealing, Third International Conference on Knowledge-Based Intelligent Information Engineering Systems, 31' Aug-1- Sept 1999, Adelaide, Australia

[8] C. Blum, M. J. Blesa Aguilera, A. Roli, and M. Sampels, editors. Hybrid Metaheuristics – An Emerging Approach to Optimization, volume 114 of Studies in Computational Intelligence. Springer, 2008.

[9] Bo Liu, PeishengMeng, Hybrid Algorithm Combining Ant Colony Algorithm with Genetic Algorithm for Continuous Domain, The 9th International Conference for Young , Computer Scientists, 2008

[10] Yi-Tung Kao, ErwieZahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, Applied Soft Computing 8 (2008) 849–857

[11] F. Focacci, F. Laburthe, and A. Lodi. Local search and constraint programming. In Glover and Kochenberger [25], pages 369–403.

[12] El-GhazaliTalbi, Metaheuristic: from design to implementation, John Wiley & Sons

[13] CristianMunteanu , Agostinho Rosa, Gray-Scale Image Enhancement as an Automatic Process Driven by Evolution, IEEE Transaction on System, Man and Cybernetics-Part B: Cybernetics, Vol. 34, No.2, April 2004

[14] J. Holland, Adaptation in Natural and Artificial systems, University of Michigan Press, Ann Anbor, 1975.

[15] J. Kennedy and R. Eberhart, Particle swarm optimization, in: Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, p. 1942-1948 (1995).

[16] R. Storn and K. Price. Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, Int CS Institute, University of California, Mar 1995.

[17] S. Kirkpatrick, C. D. Gellat, and M. P. Vecchi, Optimization by simulated annealing, Science, 220, 671-680 (1983).

[18] Nourani, Y. and Andresen, B. (1998) A comparison of simulated annealing cooling strategies. Journal of Physics A—Mathematical and General, 31, 8373–8385.

[19] Michalewicz Z. (1996) "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag.