Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów
Vol. 13, 2013, No. 1

A COMPARISON OF DIFFERENTIAL EVOLUTION, PARTICLE SWARM OPTIMIZATION, ARTIFICIAL BEE COLONY AND CUCKOO SEARCH FOR MULTILEVEL THRESHOLDING OF WASTE WOOD

SUSHIL KUMAR*, MILLIE PANT, AMIYA KUMAR RAY

Indian Institute of Technology, Roorkee
**Corresponding author: kumarsushiliitr@gmail.com*

Abstract

The present study deals with the image segmentation of waste wood material using some popular nature inspired metaheuristics like: Differential Evolution (DE), Particle Swarm Optimization(PSO) Artificial bee Colony (ABC) and Cuckoo Search (CS). Otsu's between class-variance and Kapur's maximum entropy techniques are used as fitness functions. Experiments have been performed on various images and numerical results are compared. It is observed that in some cases Otsu method is giving the same performance as DE, PSO, ABC and CS. But when class size increases DE shows better results in comparison to others.

Key words: DE, PSO, ABC, CS, thresholding

1. INTRODUCTION

Waste wood appears in many formats such as old furniture, packing crates etc. It can be reused for many purposes but a main problem is that the waste wood may contain many impurities in it like iron, plastics etc. These impurities have to be cleaned up before the waste wood can be reused. The impurities that may appear on the surface of the waste wood and inside it,are mostly invisible to the human eye. Howeverin order to make an efficient use of the waste wood, one of the option is proper detection and location of waste wood so that its removal can be done in a time effective manner.

X-Ray technology can be used for detection of impurities. First of all an X-Ray image of waste wood will be taken and is converted into a digital image by some professional image software. After that image segmentation can be applied to identify the impurities.

The concept of waste wood image segmentation is not new. Yujun et al. (2010) applied Genetic Algorithm for waste wood image segmentation. In the present study we have applied some latest algorithms Differential Evolution, Particle Swarm Optimization and Cuckoo Search for image segmentation of waste wood.

The rest of the paper is organized as follows: in the next section we give a brief introduction to the algorithms used in the present study. In Section 3, we give the image segmentation methods used in the study. Results are given in Section 4 and finally the paper concludes with Section 5. Figure 1 has taken from Yujun et al. (2010).

ISSN 1641-8581

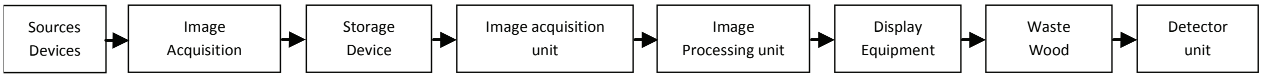

Fig. 1. *Waste Wood Material Detection System.*

2. A BRIEF INTRODUCTION TO THE ALGORITHMS USED IN THIS STUDY

2.1. Differential Evolution (DE)

Differential Evolution (DE) is a population-based, efficient, robust, and direct search method developed by Stron and Price (1995). DE works in two phase: initialization and evolution. In the first phase, a population is generated randomly for DE when no preliminary knowledge is known about the problems. In evolution phase, all the individuals go through mutation, crossover and selection process repeatedly until the termination criterion is met.

The working of DE as follows:

Initialization: Let $P_G = \{X_{i,G} | i = 1,2,\cdots,NP\}$ be the population at any generation G where each individual $X_{i,G} = \{x_{1,i,G}, x_{1,2,G}, \cdots, x_{n,i,G}\}$ is an-dimensional vector and NP is population size. Each vector of the initial population (at G=0) can be generated as given in equation-1;

Mutation: For each individual $X_{i,g}$, at any generation g, the mutant vector $M_{i,G}$ is defined by Stron and Price (1995) as given below;

$$M_{i,G} = X_{r_1,G} + F \times (X_{r_2,G} - X_{r_3,G}) \quad (1)$$

where *i* = {1, 2, ..,*NP*} and $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, different from each other and also different from the running index *i*. Since *i*, r_1, r_2 and r_3 are different so that $NP \geq 4$ is required. $F \in (0,2]$ is a real constant which controls the amplification of the difference vectors $(X_{r_2,G} - X_{r_3,G})$.

Crossover: Once the mutation phase is over, crossover is performed between the target vector and the mutated vector to generate a 'trial/offspring' vector $Y_{i,G} = \{y_{1,i,G}, y_{1,2,G}, \cdots, y_{D,i,G}\}$ and also for increase the diversity of population. It defined as follows;

$$y_{j,i,G} = \begin{cases} m_{j,i,G} & \text{if} \quad rand_j \leq Cr \vee j = k \\ x_{j,i,G} & \text{otherwise} \end{cases} \quad (2)$$

where $k \in \{1,2,\cdots,D\}$, is a random parameter index, chosen once for each *i* to make sure that at least one component is always selected from the mutated vector, $M_{i,G}$., $rand_j$ is j^{th}evolution of a uniform random number generator. Cr is predefined in Stron and Price (1995), crossover rate whose value is generally taken as $Cr \in [0,1]$

Selection: It is an approach to decide which vector ($X_{i,G}$or$Y_{i,G}$) should be a member of next generation G+1. If vector $Y_{i,G}$ yields a smaller cost function value than $X_{i,G}$ then $X_{i,G+1}$ is set to $Y_{i,G}$ otherwise, the old value $X_{i,G}$ is retained

$$X_{i,G+1} = \begin{cases} Y_{i,G} & \text{if} \quad f(Y_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (3)$$

2.2. Particle Swarm Optimization

PSO developed by Kennedy and Eberhart (1995) is a stochastic, population set based nature inspired optimization algorithm. In a PSO system, a swarm of individuals (called particles) fly through the search space. Each particle represents a candidate solution of the optimization problem. The position of a particle is influenced by the best position visited by itself and best position of a particle in its neighborhood. Suppose the position and velocity of the i^{th} particle in *n*-dimensional space at any generation *G*, are represented as $X_{i,G} = \{x_{1,i,G}, x_{2,i,G}, \cdots, x_{n,i,G})$ and $V_{i,G} = \{v_{1,i,G}, v_{2,i,G}, \cdots, v_{n,i,G})$ respectively The initial population $X_{i,G}$ (at *G*=0) generated in C.Y. Lee (2012), as below;

$$x_{j,i,0} = x_j^{\min} + rand(0,1) \times (x_j^{\max} - x_j^{\min}); j = 1,2,\cdots,D \quad (4)$$

where $x_j^{\min} \leq x_{j,0} \leq x_j^{\max}$ $\forall j$ and $rand(0,1)$ is uniform random number between 0 and 1.

The updating rules for new velocity and the position of each particle given as below:

$$V_{i,G+1} = wV_{i,G} + c_1\, rand_1\, (pbest_{i,G} - X_{i,G}) + c_2\, rand_2\, (gbest_{i,G} - X_{i,G}) \quad (5)$$
$$X_{i,G+1} = X_G + V_{i,G+1}$$

Here *w* denotes the inertia weight that is used to control the particle velocity. c_1 and c_2 are two positive numbers called acceleration constant and are usually set to 2.05, $pbest_{i,G}$ and $gbest_{i,G}$ are personal and global best position of i^{th} particle at generation

G; $rand_1$ and $rand_2$ are two uniform random numbers between 0 and 1.

Basically, the value of each component in V_i can be fix in the range $[-v_{max},v_{max}]$ to control excessive roaming of particles outside the search space.

2.3. Artificial Bee Colony

Artificial Bee Colony developed by Karaboga (2005) is based on the movement of foraging bees classified as employed, onlooker and scouts. In ABC algorithm, the solution of the problem under consideration is represented by the food source, and the quality of the solution is represented by the nectar amount of the food source.

The working of ABC is describes as follows:

Initialization: The first step of ABC is generation of initial population. We assume that the initial population consists of SN number of n-dimensional real-valued vectors (i.e., food sources) generated randomly. Suppose $X_{i,G} = \{x_{1,i,G}, x_{2,i,G}, ..., x_{n,i,G}\}$ denotes the i^th food source in the population, where each food source is initialized as given in Equation-1

Employed Bees: In this phase each employed bee x_i generates a new food source v_i in the neighbourhood of its present position by using the equation.

$$FS_{i,G} = X_{i,G} + \phi_i(X_{i,G} - X_{k,G}) \tag{6}$$

Both, I and k are randomly chosen index from 1, 2,...,NP but $k \neq i$. A tournament selection is held between $f(FS_i)$ and $f(X_i)$ and the one with a better fitness value becomes the member of the population. $\phi_i \in [-1, 1]$ is the control parameter that determines the step length of the movement of bees.

Onlookers: An onlooker bee evaluates the nectar information taken from all the employed bees and selects a food source X_i according to the probability τ_i calculated as:

$$\tau_i = f_i / \sum_{k=1}^{NP} f_k \tag{7}$$

Where f_i denotes the fitness value of I, food source (position in parameter space).

The onlooker after selecting the food source X_i, modifies it by using equation (7). Now, like the working of employed bees' phase, if the modified food source has a better or equal nectar amount than x_i, the modified food source replaces x_i and become a new member in the population. We can see that the probability of exploring the promising regions is quite high in case of onlookers

Scouts: If a food source x_i cannot be further improved after a specified number of trials limit, it signifies that the food source no longer represents a promising solution and may be abandoned. The corresponding employed bee is now termed as a scout, which produces a food source randomly with the help of (1).

2.4. Cuckoo Search (CS)

This algorithm is based on the behaviour of Cuckoo birds especially their behaviour during the laying and hatching or nursing of eggs given by Payne (2005). In the present study we have considered the CS algorithm given by Yang and Deb (2009) . This algorithm is based on the following three rules:

1) Each cuckoo lays one egg at a time, and dump its egg in randomly chosen nest.
2) The best nests with high quality of eggs will carry over to the next generations;
3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \in [0, 1]$.

In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest. For simplicity, this last assumption can be approximated by the fraction pa of the n nests are replaced by new nests (with new random solutions).

For a maximization problem, the quality or fitness of a solution can simply be proportional to the value of the objective function.

When generating new solutions $x^{(t+1)}$ for, say, a cuckoo i, a L´evy flight is performed

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \tag{8}$$

is the step size which should be related to the scales of the problem of interests. In most cases, we can use $\alpha = 1$. The above equation is essentially the stochastic equation for random walk. In general, a random walk is a Markov chain whose next status/location only depends on the current location (the first term in the above equation) and the transition probability (the second term). The product $\oplus$means entrywise multiplications. This entrywise product is similar to those used in PSO, but here the random walk via L´evy flight is more efficient in exploring the search space as its step length is much longer in the long run.

The L´evy flight essentially provides a random walk while the random step length is drawn from aL´evy Distribution

$$Levy{\sim}u = t^{-\gamma}, (1 < \gamma \leq 3) \quad (9)$$

which has an infinite variance with an infinite mean. Here the steps essentially form a random walk process with a power-law step-length distribution with a heavy tail. Some of the new solutions should be generated by L´evy walk around the best solution obtained so far, this will speed up the local search. However, a substantial fraction of the new solutions should be generated by far field randomization and whose locations should be far enough from the current best solution, this will make sure the system will not be trapped in a local optimum.

3. IMAGE SEGMENTATION

Multilevel thresholding is one of the most popular techniques for image segmentation. Image segmentation can be broadly classified as bi-level and multilevel. In bi-level image segmentation the whole image divides into partitions based on a threshold value, whereas in multilevel segmentation, as the name suggests, multiple threshold values are required. Proper threshold values should be assigned to optimise a criterion such as entropy or between-class variance, for a successful segmentation.

3.1. Image segmentation method based on Evolutionary Algorithms

1) Eight-bit binary code starting from 00000000 to 11111111 is used for representing the a gray image (0 to 255). Each binary string represents a threshold.
2) Initial population size is set to be 30.
3) Fitness Function based on maximum between-cluster variance method or OTSU method is:

Assuming an image is represented in L gray levels$[0,1,\ldots,L-1]$. The number of pixels at level i is denoted byn_i, and the total number of pixels is denoted by $n_1 + n_2 + \cdots + n_L$. The probability of gray level i is denoted by

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_0^{L-1} p_i = 1 \quad (10)$$

In the bi-level thresholding method, the pixels of image are divided into two classes C_1 , with gray levels $[0,1,\ldots,t]$and C_2,with gray levels $[t+1,\ldots L-1]$ by the threshold t. The gray level probability distributions for the two classes are given as:

$$w_1 = \Pr(C_1) = \sum_{i=0}^{t} p_i \quad (11)$$

$$w_2 = \Pr(C_2) = \sum_{i=t+1}^{L-1} p_i \quad (12)$$

The Means of class C_1 *and* C_2 are

$$u_1 = \frac{\sum_{i=0}^{t} i p_i}{w_1} \quad (13)$$

$$u_2 = \frac{\sum_{i=t+1}^{L-1} i p_i}{w_2} \quad (14)$$

The total mean of gray levels is denoted by u_T

$u_T = w_1 u_1 + w_2 u_2$ (15) (15)

The class variances are

$$\sigma_1^2 = \frac{\sum_{i=0}^{t} (i-u_1)^2 p_i}{w_1} \quad (16)$$

$$\sigma_2^2 = \frac{\sum_{i=t+1}^{L-1} (i-u_2)^2 p_i}{w_2} \quad (17)$$

The within-class variance is

$$\sigma_W^2 = \sum_{k=1}^{M} w_k \sigma_k^2 \quad (18)$$

The between-class variance is

$$\sigma_B^2 = w_1(u_1 - u_T)^2 + w_2(u_2 - u_T)^2 \quad (19)$$

The total variance of grey levels is

$$\sigma_T^2 = \sigma_W^2 + \sigma_B^2 \ (20) \quad (20)$$

Otsu method chooses the optimal threshold t by maximizing the between-class variance, which is equivalent to minimizing the within-class variance, since the total variance (the sum of the within-class variance and the between-class variance) is constant for different partitions. Objective function is:

$$t = \arg\left\{max_{0 \leq t \leq L-1}\{\sigma_B^2(t)\}\right\} = \arg\left\{min_{0 \leq t \leq L-1}\{\sigma_W^2(t)\}\right\} \quad (21)$$

3.2. Entropy based thresholding technique

Entropy-based thresholding segmentation technique is based on the probability distribution of the gray level histogram. The purpose is to find the optimal thresholds yielding the maximum entropy because when entropy is maximum the optimal thresholds separating the classes are assigned properly. The entropy of a discrete source is obtained from the probability distribution $p = p_i$, where p_i is the probability of the system in possible state i. The probability of each gray level i is the relative occurrence frequency of the gray level i, normalized by the total number of gray levels as described in equation:

$$p_i = \frac{h(i)}{\sum_{i=0}^{L-1} h(i)}, i = 0,1,2, \dots, L-1 \quad (22)$$

For bi-level thresholding, Kapur's entropy may be described by equation:

$$H_0 = \sum_{i=0}^{t-1} \frac{p_i}{\omega_0} ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t-1} p_i \quad (23)$$

$$H_1 = \sum_{i=t}^{L-1} \frac{p_i}{\omega_1} ln \frac{p_i}{\omega_1}, \omega_1 = \sum_{i=t}^{L-1} p_i \quad (24)$$

The threshold is optimum when the summation of the class entropies are maximumas described in given equation, it is objective function:

$$t^* = \arg\max(H_0 + H_1) \quad (25)$$

For multilevel thresholding Kapur's entropy can be extended as described in given equation.

$$H_0 = \sum_{i=0}^{t_1-1} \frac{p_i}{\omega_0} ln \frac{p_i}{\omega_0}, \omega_0 = \sum_{i=0}^{t_1-1} p_i \quad (26)$$

$$H_1 = \sum_{i=t_1}^{t_2-1} \frac{p_i}{\omega_1} ln \frac{p_i}{\omega_1}, \omega_1 = \sum_{i=t_1}^{t_2-1} p_i \quad (27)$$

$$H_2 = \sum_{i=t_2}^{t_3-1} \frac{p_i}{\omega_2} ln \frac{p_i}{\omega_1}, \omega_2 = \sum_{i=t_2}^{t_3-1} p_i \quad (28)$$

$$\dots$$

$$H_c = \sum_{i=t_c}^{L-1} \frac{p_i}{\omega_c} ln \frac{p_i}{\omega_c}, \omega_c = \sum_{i=t_c}^{L-1} p_i \quad (29)$$

The multilevel thresholding consists C dimensional vector $T = \{(t_1, t_2, \dots, t_c): t_1 < t_2 <, \dots < t_c\}$, which optimizes the objective function:

$$t^* = \arg\max(H_0 + H_1 + H_2 + \dots + H_c) \quad (30)$$

4. EXPERIMENTAL SETTINGS AND BENCHMARK PROBLEMS

In the present study we considered 3 gray scale images, named figure 2, figure 3, figure 4. Experimental settings are presented in table 1.

Table 1. *Parameter settings for different algorithms.*

Pop size (*NP*) 1 to *L*
Max NFE 10^5
Total Run 1
Differential Evolution
Scale Factor (*F*) - .5,
Crossover rate (*Cr*)-0.9
Particle Swarm Optimization
Inertia weight (*w*) - 0.5
*c1*and *c2* - 2.0
Artificial Bee Colony
Max Cycle Number (MCN) 1000
Limit 100
Cuckoo Search
pa = 0.25

5. EXPERIMENTAL RESULTS AND ANALYSIS

Results in this section shows a significant difference among all these evolutionary algorithms like PSO, ABC, CS and DE with Otsu segmentation method. Threshold difference among all these algorithms shows that differential evolution performs very well for threshold selection.

Fig. 2.

Fig. 3.

Fig. 4.

6. CONCLUSIONS

Image segmentation can be very useful for the reuse of waste wood. In the present study we used some popular metaheuristics for image segmentation of waste wood. Objective functions used are by Otsu and by Kapur's entropy method. It was observed that all the algorithms performed well for both the objective functions with DE giving the best results when the class size is increased.

Table 2. Comparison among PSO, ABC, CS, DE and Otsu method for between class variance and threshold selection.

	Between Class variance						Thresholds				
m	**figure 1**										
	Otsu	PSO	ABC	CS	DE		Otsu	PSO	ABC	CS	DE
2	.942621	.942621	.942621	.942621	.942621		89, 167	89, 167	89, 167	89, 167	87 ,169
3	.943515	.943624	.944201	.948931	.948456		74, 151, 187	73, 155, 194	73, 149, 198	75, 147, 197	74, 154, 196
4	.948359	.949732	.955712	.958719	.958177		67, 95, 156, 200	72, 107, 154, 190	75, 102, 188, 208	74, 105, 189, 199	76, 99, 196, 204
5	.953763	.954216	.959354	.952563	.959821		57, 95, 121, 155, 207	49, 81, 110, 158, 201	57, 85, 120, 158, 201	59, 78, 107, 150, 207	62, 85, 121, 158, 207
	figure 2										
2	.924319	.924319	.924319	.924319	.924319		74, 156	74, 156	74, 156	74, 156	72, 153
3	.943652	.943731	.945783	.946198	.945937		66, 112, 169	64, 116, 161	66, 115, 193	64, 114, 190	69,124,168
4	.952538	.953272	.953875	.954892	.945671		67, 98, 126, 176	32, 69, 119, 184	55, 97, 131, 167	47, 97, 111, 189	49, 99, 171, 185
5	.964533	.965328	.965382	.967862	.966819		48, 97, 134, 152, 197	28, 85, 120, 155, 170	59, 77, 117, 173, 168	45, 83, 105, 174, 200	45, 79, 120, 165, 212
	figure 3										
2	.924753	.924753	.924753	.924753	.924753		88, 173	88, 173	88, 173	88, 173	88, 173
3	.937184	.938342	.938473	.938762	.939875		78, 159, 210	76, 158, 201	80, 146, 194	82, 134, 201	83, 143, 197
4	.946135	.947218	.948548	.949761	.949659		49, 99, 149, 200	51, 89, 121, 196	65, 85, 108, 189	51, 86, 111, 201	54, 87, 132, 196
5	.956208	.957291	.958632	.959624	.959866		48, 88, 132, 169, 213	61, 73, 111, 155, 205	38, 74, 101, 152, 197	28, 70, 99, 152, 198	58, 85, 102, 137, 211

REFERENCES

Benala, T.R., Jampala, S.D., Villa, S.H., Konathala, B., 2009, A novel approach to image edge enhancement using artificial bee colony optimization algorithm for hybridized smoothening filters, *World Congress On Nature And Biologically Inspired Computing (NABIC 2009)*, 1070-1075.

Becerra, R.L. Coello, C.A., 2004, Culturizing differential evolution for constrained optimization, *Proc. Conf. 5. Mexican International Conference in Computer Science,*304-311.

Kennedy, J., Eberhart, R.C., 1995, Particle swarm optimization, *IEEE International Conference on Neural Networks,* 4, 1942-1948.

Karaboga, D., 2005, An idea based on honey bee swarm for numerical optimization, *Technical Report TR06*, Erciyes University.

Lee, C.Y., Leou, J.J., Hsiao, H.H., 2012, Saliency-directedcolor image segmentation usingmodified particle swarm optimization, *Signal Processing,* 92, 1-18.

Price, K., Storn, R., Lampinen, J.A., 2005, *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*, Springer.

Price, K., 1999, *An Introduction to differential evolution, New Ideas in Optimization*, McGraw-Hill, London (UK).

Payne, R. B., Sorenson, M. D., Klitz, K., 2005, *The Cuckoos*, Oxford University Press.

Price, K., 1999, *An introduction to differential evolution,* New Ideas in Optimization, McGraw Hill, London, 79-108.

Price, K., 1996, Differential evolution: a fast and simple numerical optimizer, Biennial Conference of the North American Fuzzy Information Processing Society, IEEE Press, New York, 524-527.

Price, K.V., 1997, Differential evolution vs. the functions of the 2ndICEO, *Proc. IEEE International Conference on EvolutionaryComputation*, 153-157.

Storn, R., Price, K., 1997, Differential Evolution—A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces, *J Glob Optim,* 11, 341-359.

Storn, R., Price, K., 1995, Differential Evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical Report TR-95-012.*

Yujun, Z., Jiangming, K., Jinhao, L., Derong, Z., 2010, The Application of Genetic Algorithm for the Segmentation of Measured Image of Waste Wood Material Connectors, *Proc. International Conference on Computing, Control and Industrial Engineering,* 170-174.

Yang X.S., Deb S., 2009, Cuckoo search via L´evy flights, *Proc. Of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, 210-214.

PORÓWNANIE METODY EWOLUCJI RÓŻNICOWEJ, METODY ROJU CZĄSTEK, ALGORYTMU PSZCZELEGO I ALGORYTMU KUKUŁKI DO WIELOWARIANTOWEJ KLASYFIKACJI ODPADÓW DREWNA

Streszczenie

Przedstawione badania dotyczą segmentacji obrazów odpadów drewna przy użyciu popularnych algorytmów inspirowanych naturą, takich jak: metoda ewolucji różnicowej (DE), metoda roju cząstek (PSO), algorytmu pszczelego (ABC) oraz algorytmu kukułki (CS). Jako funkcję celu wykorzystano wariancję międzyklasową Otsu oraz zasadę maksymalnej entropii. Porównując wyniki otrzymane dla różnych obrazów, zaobserwowano, że w niektórych przypadkach metoda Otsu wykazuje taką samą wydajność jak DE, PSO, ABC i CS. Jednak przy wzroście liczby klas wyniki otrzymane metodą DE są lepsze niż otrzymane pozostałymi metodami.

Received: September 20, 2012
Received in a revised form: October 8, 2012
Accepted: November 27, 2012