Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial
ISSN: 1137-3601
editor@iberamia.org
Asociación Española para la Inteligencia Artificial
España

Loubna, Benchikhi; Mohamed, Sadgal; Abdelaziz, Elfazziki; Fatimaezzahra, Mansouri
A Novel adaptive Discrete Cuckoo Search Algorithm for parameter optimization in computer vision
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 20, núm. 60, 2017, pp. 51-71
Asociación Española para la Inteligencia Artificial
Valencia, España

Inteligencia Artificial 20(60) (2017), 51-71
doi: 10.4114/intartif.vol20iss60pp51-71

INTELIGENCIA ARTIFICIAL

http://journal.iberamia.org/

A Novel adaptive Discrete Cuckoo Search Algorithm for parameter optimization in computer vision

Benchikhi Loubna[1], Sadgal Mohamed[1], Elfazziki Abdelaziz[1], Mansouri Fatimaezzahra[1]

[1] Cadi Ayyad University, Laboratoire d'Ingénierie des Systèmes d'Information
Marrakesh, Morocco
l.benchikhi@uca.ma
sadgal@hotmail.com
elfazziki@uca.ma
fatimaezzahra.mansouri@ced.uca.ma

Abstract Computer vision applications require choosing operators and their parameters, in order to provide the best outcomes. Often, the users quarry on expert knowledge and must experiment many combinations to find manually the best one. As performance, time and accuracy are important, it is necessary to automate parameter optimization at least for crucial operators. In this paper, a novel approach based on an adaptive discrete cuckoo search algorithm (ADCS) is proposed. It automates the process of algorithms' setting and provides optimal parameters for vision applications. This work reconsiders a discretization problem to adapt the cuckoo search algorithm and presents the procedure of parameter optimization. Some experiments on real examples and comparisons to other metaheuristic-based approaches: particle swarm optimization (PSO), reinforcement learning (RL) and ant colony optimization (ACO) show the efficiency of this novel method.

Keywords Computer vision; image processing; parameter optimization; metaheuristic; ADCS; quality control.

1 Introduction

Quality Control based on computer vision requires a very high precision in object extraction often under imposed constraints. A variety of methods have been developed since decades towards image processing in quality control, but the problem of selecting vision operators and the adjustment of their parameters is still relevant. Parameters tuning is usually made by trial and error; algorithms may be adopted after a long series of tests. However, it is a complex and time-consuming task due to the unclear and complex relationship between input parameters and outputs.

1.1 Problematic

The choice of operators and their parameters require specific knowledge, and must be done experimentally due to the lack of automatic mechanisms. In spite of vision systems' diversity and the rich library of image processing methods, the user intervention in most applications remain necessary. Few systems succeeded in automating vision applications without requiring user's intuition. In several studies, authors propose methods such as numerical optimization, which applies mathematical or statistical techniques to optimize an objective function defined over a parameter space.

In general, operators use either quantitative variables, which may change according to a continuous scale, or qualitative ones which can only assume certain discrete values. If numerical optimization is used for continuous variables, it presents a problem for qualitative ones.

ISSN: 1137-3601 (print), 1988-3064 (on-line)

Another issue related to image processing is that outcomes' quality is not directly expressed with parameters (in a mathematical way). This makes the use of mathematical models (formal methods) very difficult if not impossible. To meet this difficulty, specialists are moving towards learning and optimization through metaheuristics. Thus, to fix parameters, some experiments seem necessary. Prior to conducting any experiments, the experimenter has to specify some input conditions: variables number (controllable ones and uncontrollable), their range, number of responses and the experience objective.

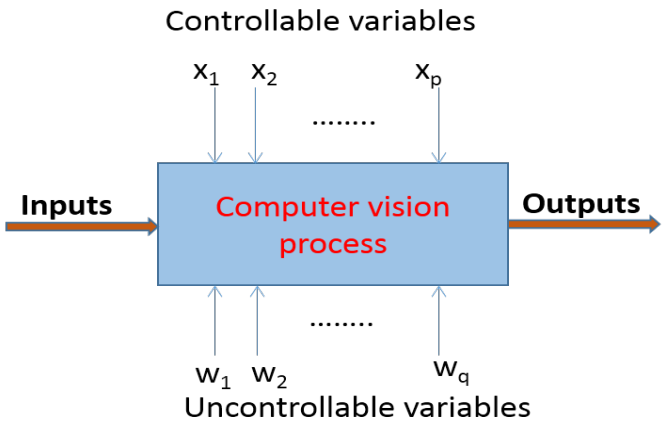


Figure 1. Design of experiments for computer vision applications

Each computer vision process contains inputs in addition to variables (such as thresholds, the number of classes, filter size, etc.), some of these variables can be monitored and some others cannot, the experimenter must insist on variables that affect the outcomes. Thus, the model can be formed in an iterative way or through distribution by, test and error. Most of the proposed techniques for parameter tuning have not been widely adopted. The problem has been solved using learning models [1] and optimization models [2][3][4][5]. This can be partly awarded to few application examples in the real image analysis. Visually, with different parameter types and the variety of their values, the problem is NP-complex.

1.2 Contribution and outline

Our objective is to deploy the artificial intelligence techniques [6] and optimization metaheuristics for solving the problem of operators and their parameters tuning. The most important property in mataheuristics is the cooperation between individuals to provide the best solution. Therefore, metaheuristics based on population methods can accomplish this role to explore and construct solutions.

The contribution of this paper is the proposition of a novel adaptive discrete model based on CSA (Cuckoo Search Algorithm) adapted to parameter adjustment. In most cases, vision operators use discrete domains of their settings, which require an adaptation of the continuous methods. The standard CSA is performed for continuous problems, thus, we focus in this work on the problem reformulation in a discrete way to avoid the continuity constraint and to provide a modified algorithm.

The choice of cuckoo search algorithm is based on different reasons:

1. CSA is a population-based metaheuristic. To explore the search space, the algorithm uses random walk via Lévy flight [7], which is very efficient in discovering important solutions.

2. As other metaheuristics PSO, ACO and GA (genetic algorithm), the CSA has some parameters that must be adjusted, but their number remains less important compared to other algorithms. CSA is potentially more suitable for a wide class of optimization procedures.

The cuckoo search algorithm proposed by Yang and Deb [8] is based on random walk models such as Lévy flight and bird parasitic behavior. CSA succeeds perfectly in optimizing continuous functions [9]; it is now revised

to solve discrete problems, case by case, such as knapsack problem [10] and nurse scheduling problem [11]. In the case of optimal parameter values for applications in vision systems, they are found using a variant of CS.
In this paper, we try to demonstrate three points:

- Formulation of the parameter tuning problem in vision tasks. The computer vision application can be considered as a project management; so a task will be presented as a PERT Chart (Fig. 2): phases (nodes) are linked by operators (arcs) according to succession constraints in the vision process. This Description generalizes a vision task presentation that can be used by any metaheuristic.
- The stepsize adaptation: vision task may take a long time because of a large number of iterations and operators' complexity. For applications in quality control, it is essential to reduce the convergence rate and save execution time. We proposed a modified Levy's stepsize by introducing iteration number, called also generation time.
- Problem discretization: we propose a discrete method to adapt the continuous random walk functions (like Lévy flight) in order to use discrete inputs.

Using experimentation and multiple comparisons, it is shown that the adaptive discrete cuckoo search algorithm (ADCS) is more adapted for parameter optimization and allows achieving good quality results. The approach is applied on real image examples and outputs are compared to references (ground truth).

1.3 Paper organization

The remainder of this paper is organized as follows: Section 2 gives an overview of related work. In section 3, the problem of parameter optimization relatively to the vision tasks is formulated. Section 4 describes the proposed approach. The experimental results are presented in Section 5 and the conclusion is stated in Section 6.

2 Related work

In the majority of vision tasks, the user is required and sometimes obliged to combine several operators, each one has a multitude of parameters to adjust, but few systems have succeeded in automating vision applications without requiring user's intuition.

As first works, we can cite Taylor [1] who proposed a method based on reinforcement learning to monitor parameters in vision applications. B.Nikolay and al. [12] proposed a method to optimize automatically parameters of vision systems for surface inspection. This method is based on evolutionary algorithms. The two major types used are evolutionary strategies (ESs) and evolutionary programming (EPs). However, different techniques have been developed on the basis of metaheuristics and especially population approaches. Some authors searched to automate completely the process, S.Treuillet and al. [13] proposed a method to adjust parameters in an image processing chain based on an experimental 2^{k-p} factorial plan applied to a vision system designed for measuring the neck ratio of a sugar beet batch. Recently L.Franek and X.jiang [14] proposed to use orthogonal design of experiments for parameter learning. Means are analyzed to estimate the optimal parameter setting. To further improve the performance, a combination of orthogonal arrays and genetic algorithm is used. In recent years, parameter optimization in image processing is supported by artificial intelligence techniques, such as multi-agent architecture. I.Qaffo and al. [6] proposed an automatic method based on reinforcement learning for object recognition. They proposed agents to provide a combination of operators and agents to optimize parameters on the basis of Q-learning to extract object of interest from images.
Although several exact algorithms such as dynamic programming, branch-and-bound and linear programming, shown their efficiency to solve parameter setting problem, some metaheuristics showed its potential performance as well. In contrast, to early heuristic, a population-based metaheuristic uses multiple solutions at the time. Exploring the solution space (or search space) within multiple locations at the time increases diversity and makes these algorithms more robust compared to trajectory-based ones [15]. Many population-based metaheuristics proposed to solve the optimization problems: Genetic Algorithms [16], Ant Colony Optimization (ACO) [17] and others as binary glowworm swarm optimization algorithm [18] or Cuckoo Search [19] and Firefly Algorithm [20], etc. In such approaches, each individual (or agent) in a population starts by building an approximate solution. With a mechanism of interaction and evolution, individuals converge toward the optimal solution.
The CSA provides best results for parameters' tuning in some applications where the result is expressed directly with parameters in continuous functions [21]. Works of Yang and Deb [8], Civicioglu and Besdok [22], Rajabioun [23], Valian and al. [24] further confirmed that the Cuckoo Search algorithm, in its original and improved version, proves to be very efficient. The method has been tested on many benchmark functions of varied dimensions and difficulty levels, and showed its success.

Since the first appearance of cuckoo search in 2009, we can enumerate some variants developed by many researchers, discrete variants are limited to few problems like knapsack problem and Transport Salesman Problem (TSP) [25]. But there is no general CS discretization.

3 Formulation of the parameter optimization problem

3.1 Vision system and tasks

Vision systems are designed to produce applications in image processing, object recognition, etc. Such application comes with several tasks and operators that transform a product stream into an information flow. An application consists of a succession of tasks (a task can be a set of other sub-tasks or an elementary operator), some tasks can be executed in parallel, over phases of treatment. It can be presented as a flow chart (Fig. 2).

In the quality control domain, vision application consists of supervising the basic parameters describing image quality, in order to extract necessary measurements to be controlled. Here, the main problem is segmentation quality that affects hardly outcomes. Image processing helps increasing flexibility and productivity in product manufacturers. Further, it aids in maintenance and enhances knowledge about products' quality. Furthermore, it offers the advantage to interfere at several levels, as the real-time quality inspection of gelatin capsules in pharmaceutical applications [26].

3.2 Operators and parameters

To accomplish a vision application, it is necessary to go through multiple tasks; each one is a succession of several phases (states in the PERT chart Fig. 2). Each phase has a set of possible operators. A phase can be formed of one or several operators.

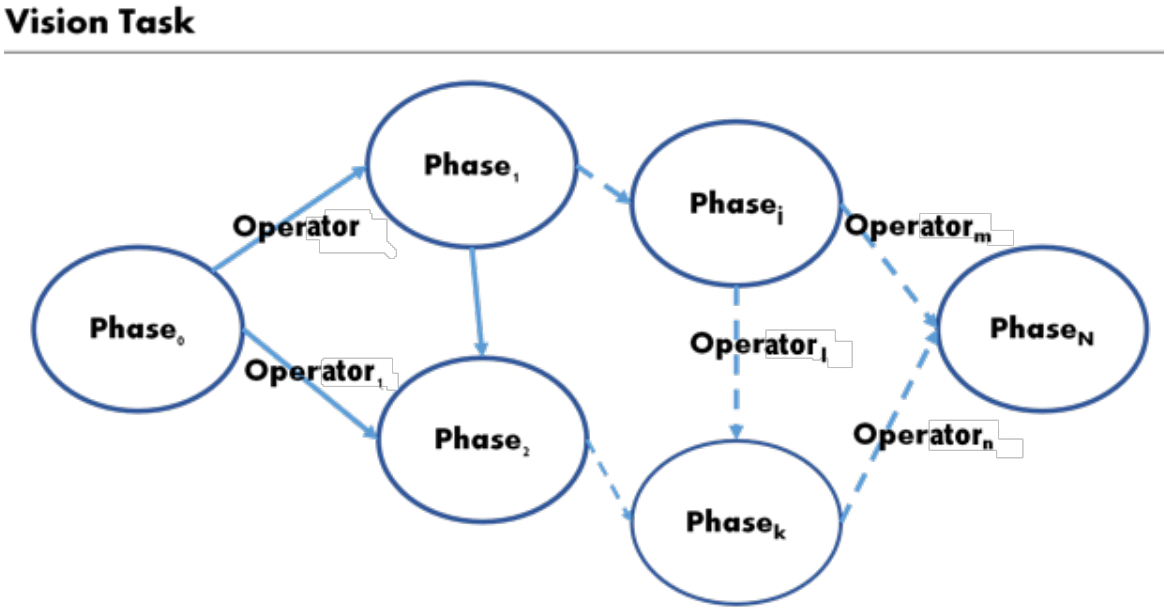


Figure 2. Possible operators for each phase in a vision task

We may have several combinations of these operators, each combination can achieve the considered task, but the output is qualitatively different. Let's consider a task made of several phases. Since each phase owns a set of feasible operators, n different operators' combination C_k can be build.

$$C_k = (O^k_1, O^k_2, ..., O^k_N) \qquad k=1, ..., n. \tag{1}$$

N represents operators' number.

An operator $\boldsymbol{O^k_j}$ in a combination $\boldsymbol{C_k}$ requires fixing some parameters (p_{j1}, …, p_{jMj}); a range of possible values for

each parameter is given out (Fig. 3).

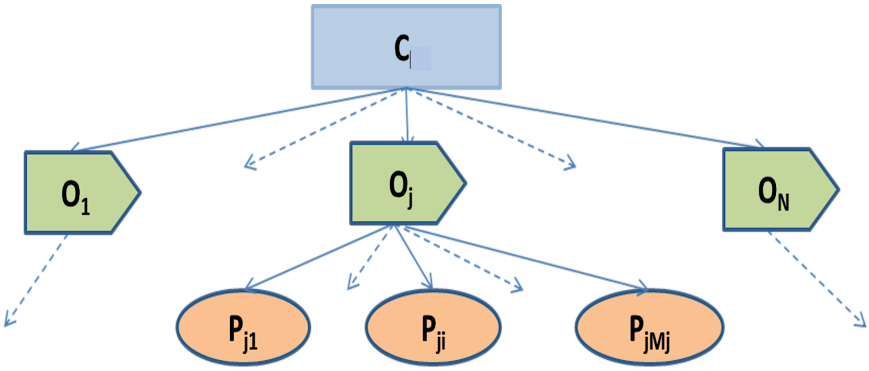


Figure 3. An Operators' combination and their parameter

The vision task performance is based on these parameter settings, and their variation strongly influences results. Below we mention some adjustable parameters of different algorithms such as edge detection. When applying an edge operator, multiple filters can be applied such as Canny, Sobel, Prewitt , …, each one has a free parameter: the threshold, to remove edges with poor contrast and then contours are formed by pixels higher than a given threshold. Fig. 4 illustrates contours done with different threshold values for a chosen filter, "canny".

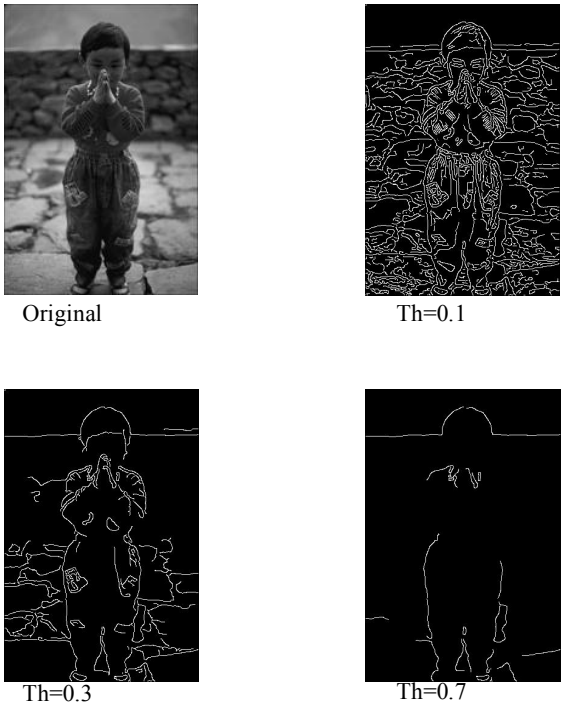


Figure 4. Canny filter operator with different threshold (Th) values

3.3 Objective function (fitness)

Mathematically, our problem can be solved by using a model $P = (D, \omega, Err)$ where:

- D is the solution Space, defined over a set of variables, and ω a set of constraints among the variables.
- An objective function Err: $D \rightarrow R^{+}$ to be minimized.

A solution $d \in D$ is a complete assignment in which each variable has an assigned value; d must satisfy all the constraints. A feasible solution d^* is a global optimum if:

$$Err(d^*) < Err(d) \quad \forall d \in D \tag{2}$$

Let's consider C_k a combination of operators and an input image I_{input}. The application of C_k provides an output result R_k (images or features); we can write:

$$R_k = C_k(I_{input}) \tag{3}$$

To evaluate the result, an objective function is necessary to compare quality of the outputs (R_k) to a reference $R_{reference}$ led by an expert and used as a ground truth. The rating calculated by this objective function represents the error between the two results.

$$Err(C_k) = fitness(R_k, R_{reference}) \tag{4}$$

The fitness function definition depends on the kind of the vision task.

For sure optimal parameter values give the best output, we have to find a model to minimize *Err* in the solution space.

If all combinations are used, we retain the best one. Consequently, this corresponds to the smallest error:

$$C_{optimal} = \underset{k=1 \ldots n}{Arg(Min(Err(C_k)))} \tag{5}$$

3.4 State definition

Let's consider $C=(O_1, O_2, \ldots, O_N)$ a combination of operators applied to the input image I_{input}, based on ''Fig. 5'' we note:

$$I_{output} = C(I_{input}) = (O_1, O_2, \ldots, O_N)(I_{input}) \tag{6}$$

I_{output} is the result of combination C application on the input image I_{input}. Considering that each operator O_j has m_j parameters and $m = \sum_{j=1}^{N} m_j$ is the number of all parameters. For a specific operator's combination the output depends on parameters:

$$I_{output} = C[p_1, \ldots, p_m](I_{input}) \tag{7}$$

So, the error is simply function of parameters denoted by:

$$Err(p_1, p_2, \ldots, p_m) = fitness(I_{output}, R_{reference}) \tag{8}$$

Where p_i is a parameter taking values in a domain D_i. Then, we consider the Cartesian product $D=D_1*D_2*\ldots*D_m$, and we call a state of parameters' values each $(u_1, u_2, \ldots, u_m) \in D$ where u_i is a value of parameter p_i. Using these notations, the problem is to find a state that minimizes the error function over the domain D. The solution is the state $(u_1^*, u_2^*, \ldots, u_m^*) \in D$, such as $Err(u_1^*, u_2^*, \ldots, u_m^*)$ is minimal:

$$(u_1^*, u_2^*, \ldots, u_m^*) = \underset{(u_1, u_2, \ldots, u_m) \in D}{ArgMin(Err(u_1, u_2, \ldots, u_m))} \tag{9}$$

The objective function is not expressed directly with parameters, but it is established on the basis of results. In fact, there is no mathematical model to be applied and the direct formal methods could not be carried out to solve this problem. In fact, as already mentioned, there are two difficulties: the variables could be numerical (continuous or not), categories, etc. In addition to that, there is no model linking variables to results. The proposed method belongs to relaxation ones, which are preferred to solve this kind of problem. Our approach consists of searching to converge toward minimal error in an iterative way, relying on an optimization model.

3.5 Resolution

The parameter optimization is classified as an NP-difficult problem. The need to find good solutions promotes the emergence of rough or stochastic algorithms namely metaheuristics [27].
In this context, metaheuristics showed their performance for a wide variety of optimization problems. These methods have more advantages compared to other algorithms, such as the ability to handle very high levels of complexity, the ability to adapt to several issues and their application in many domains, starting from operational research to artificial intelligence [28].
Metaheuristics use strategic research to explore the solution space and often focus on promising areas. These methods start with an initial set of solutions (i.e. starting population), then, other solutions are examined step by step to find out the optimal one. The two most important advantages are simplicity and flexibility; These algorithms are very flexible and have the ability to treat problems with objective functions of various properties, whether continuous, discrete or mixed.
Following, we describe a procedure employed to solve the parameter adjustment problem; it is a metaheuristic applying a variant of CSA: The adaptive discrete cuckoo search algorithm (ADCS).

4 The proposed approach

4.1 Methodology

Let's define p_1, p_2, …, p_n as a set of parameters belonging to a vision task. To evaluate a vision task, a metric is defined to measure qualitatively how similar a vision task outcome is to a reference output, more the metric's value is small, more the vision task outcome is similar to the reference output. Thus, it is necessary to find the set of parameter values related to the minimum value of this metric, which will be called p^* according to Eq. 9.

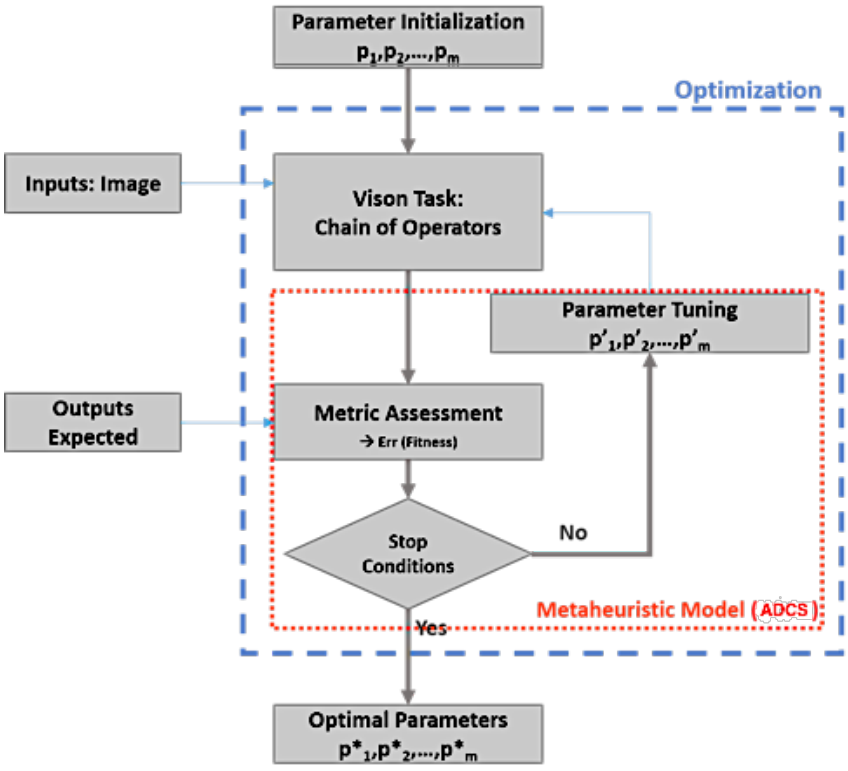


Figure 5. Optimization procedure

Fig. 5 exposes general methodology for the parameter optimization problem. First, we define an operator's combination and their parameters. Then, the procedure runs iteratively applying operators on input image, and calculates outputs which are compared to those expected. If stop conditions are not satisfied, the metaheuristic propose new parameter values to be used in the next iteration. Many metaheuristics could be used; the choice depends on results quality and response time. In the best case, the metaheuristic provides optimal parameters by convergence. In this work the ADCS is applied.

4.2 Cuckoos based heuristic

Cuckoo optimization is based on real life of a bird called cuckoo. The basis of this novel optimization is specific breeding and egg laying. Only adult cuckoos and their eggs are used in this modeling. Adult cuckoos lay eggs in other birds' habitat. Those eggs grow and become a mature cuckoo if they are not discovered and removed by host birds. Cuckoos are fascinating birds because of their aggressive reproduction strategy.
There are two algorithms cuckoo based applied for optimization: cuckoo search algorithm (CSA) and cuckoo optimization algorithm (COA).
CSA was proposed by Yang and Deb [7, 8] to optimize continuous functions. Under this work, Rajabioun in [23] has developed COA by introducing cuckoo groups.

4.2.1 Cuckoo search algorithm

We use three idealized rules to simplify description of our new cuckoo search:

1) Each cuckoo puts an egg at a time, and places its egg in a randomly chosen nest.
2) High quality eggs (in best nests) will carry on to the next generation.
3) A fixed number of host nests, $p_a \in [0, 1]$ represent the probability of an egg to be discovered by the host bird. In this case, the host bird can either throw the egg away or abandon the nest.

The fraction p_a of the n nests replaced by new nests with new random solutions, can approximate the last assumption. For a minimization problem, the quality (or fitness) of a solution is proportional to the objective function value. Based on the three rules above, the basic steps of cuckoo search algorithm are summarized in the pseudo code shown in Fig. 6.

```
Generate initial population of n host nests
while (t < MaxGeneration or stopCriterionNotFulfilled)
    Generate a solution by Lévy flights
    and then evaluate its quality/fitness Fi
    Choose a nest among n (say, j) randomly
    if (Fi < Fj),
        Replace j by the new solution
    end
    Abandon the pa nests with worst quality and build new ones
    Keep best solutions/nests
    Rank the solutions/nest and find the current best
    Pass the current best to the next generation.
end while
Postprocess results and visualization
```

Figure 6. Pseudo code for Cuckoo Search with Lévy flight

When generating a new solutions X(t+1), for a cuckoo i, a Lévy flight is performed:

$$X(t+1)_i = X(t)_i + \alpha \otimes \text{Lévy}(\beta) \quad (10)$$

Where $\alpha > 0$ is the step size, which should be related to the problem of interests. In most cases, we can use $\alpha = 1$. Equation above represents the stochastic equation for random walk. In general, a random walk is a Markov chain, whose next status (or location) only depends on the current location, represented by the first term in the above equation, and the transition probability represented by the second term.

$$\text{Lévy}(\beta) \sim |u|^{-\beta}, \ 1 < \beta \leq 3, \quad (11)$$

Where $1 < \beta \leq 3$ is an index.

A new position X(t+1) in CSA is updated using (10). Lévy flight generation of distributed numbers can be achieved following two steps:

1. The choice of a random direction.
2. Generation of steps should obey the chosen Lévy distribution.

4.2.2 The stepsize Lévy-based model

Either Mantegna algorithm or McCulloch's algorithm gives the Levy distribution, which allows to get the Levy step. Some authors suggested that Levy distribution using McCulloch's algorithm is more potent than the Mantegna's algorithm. But, the easiest way [29] is Mantegna algorithm for a symmetric Lévy stable distribution. The step length Lévy(s) [30] can be calculated using:

$$Lévy(s) = \frac{u}{|v|^{1/\beta}} \tag{12}$$

Where u and v are drawn from normal distribution:
$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2)$

$$\sigma_u = \left[\frac{\Gamma(1+\beta)\sin(\Pi\beta/2)}{\Gamma\left[\frac{1+\beta}{2}\right]\beta 2^{\frac{\beta-1}{2}}}\right]^{1/\beta}, \ \sigma_v = 1 \tag{13}$$

Γ is the Gamma function and the parameter β in Lévy distribution has different values according to different instances. In general, $1 < \beta < 3$.
The following equation uses a variance to calculate step size of random walk:

$$stepsize = (l * (\frac{u}{|v|^{\frac{1}{\beta}}}) \otimes (X(t)_i - X_{best})) \tag{14}$$

Where l= 0.01 is a factor for controlling cuckoo steps (in fact, $l=L/100$ where $L=MaxStepLength$), $\boldsymbol{X(t)_i}$ is the current solution of cuckoo i at iteration t, $\boldsymbol{X_{best}}$ is the global best solution. Then, the update equation will be:

$$X(t+1)_i = X(t)_i + stepsize \tag{15}$$

4.2.3 Step size adaptation

According to the standard version of CSA, α in Eq. 10 or α=l in Eq. 14 that handle local and global search, are given as application-dependent constants. In vision tasks, the variability of results depends not only on the operators' complexity, but also on the processed images type. Instead of trying to fix a nominal value for each task and each type of images, we propose to adjust automatically this value according to the number of elapsed generations and the fitness value of the current generation. The adapted CSA step size is illustrated by:

$$stepsize_i(t) = \alpha_t * \left(\frac{u}{|v|^{\frac{1}{\beta}}}\right) \tag{16}$$

Where:

$$\alpha_t = \frac{1}{t^{1/2}} * \frac{(BestFitness - Fitness(Xi))}{(BestFitness - WorstFirness)}$$

t = the generation number.
WorstFitness = the worst fitness values of all nests.
BestFitness = the best fitness values of all nests.

The stepsize is initially high, but when generations increase the step size decreases. Eq. (16) clearly indicates that the step size decides in an adaptive way according to fitness values.
Then the adaptive cuckoo search algorithm (ACS) is modeled by:

$$X(t+1)_i = X(t)_i + randn \times stepsize_i\,(t+1) \tag{17}$$

randn represent a random number in]0,1].

4.2.4 The proposed adaptive discrete cuckoo search

In the case of image processing, parameter domains are discrete and some are non-digital. Obviously, we cannot use directly the CSA standard version designed for continuous domains. Several studies on particular applications worked on discrete formulations of CSA designed for particular problems like TSP [30].

Based on this idea, our approach defines a domain as a set of states with an order on objective function values. We also present some operations with positions and random walk function such as addition and subtraction. By these concepts, we propose a novel adaptive discrete CSA model based on following definitions:

- $D=D_1*D_2*...*D_M$ as a set of states (search space).

  Each M-uplet $X=(u_1, u_2, ..., u_M)\in D$ is a state or a cuckoo egg.
- Objective function Err is discrete and numerical with an order on states:

  $Err(X)>= Err(X')$ or $Err(X') >= Err(X)$
- Cuckoos' Eggs represent states
- Domain representation: each value in the domain can be represented by its relative location.

The model is illustrated in Fig. 7

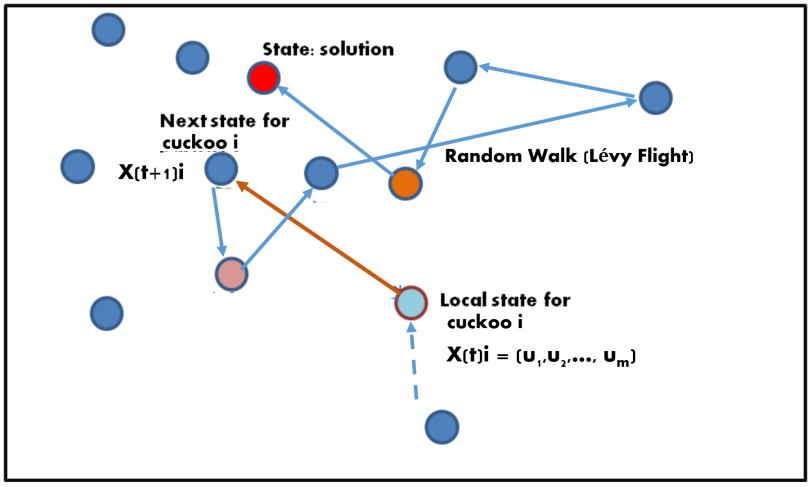


Figure 7. Space State Presentation

A domain representation can be expressed in many ways to facilitate the operator's definition on states. The representation adopted in this work is given below:

A domain $D_j=\{u_{1j},...,u_{kj},..., u_{nj}\}$ is ordered from one to $n_j=|D_j|$. Each value has a location in the domain and then two reciprocal functions *Rank* and *Value are* defined:

$Rank(u_{kj})=k$, rank of the value u_{kj} in D_j.

$u_{kj} = Value(k)$, represents the value in D_j corresponding to the rank k.

Example:

$D_j=\{a, b, c, d\}$

$Rank(a)=1, ..., Rank(d)=4$

$Value(1)=a, ...,Value(4)=d$

Since the Lévy flight uses numerical operations, the values of a non numerical parameter will be represented by their ranks.

4.2.5 The update functions

Since the ADCS algorithm claims to define positions discretely and updates them using random walk, we describe below our proposition.

a) Addition and subtraction operators

To add two values at rank k and k' we move to the rank $k+k'$ and the result is the value corresponding to rank $k+k'$. Since D_j has a limited size $|D_j|$, $k+k'$ is calculated modulo $|D_j|$. Then, the addition operator is defined as:

$u_{kj} \oplus u_{k'j} = u_{(k+k')j}$

Idem for Subtraction, the result corresponds to $k-k'$ modulo $|D_j|$:

$u_{kj} \ominus u_{k'j} = u_{(k-k')j}$

A cuckoo position is a state $X = (u_1,u_2,..u_M)$, were u_j are values in D_j. To change the state, operations are extended to states:

$X \oplus X' = (u_1 \oplus u'_1, u_2 \oplus u'_2, .., u_M \oplus u'_M)$,

$X \ominus X' = (u_1 \ominus u'_1, u_2 \ominus u'_2, .., u_M \ominus u'_M)$,

$Rank(X) = (Rank(u_1), \ldots, Rank(u_M))$,

$X = (Value(k_1), \ldots, Value(k_M))$, were k_j is a rank of a value in D_j.

b) Rank update: Discretization of Levy's stepsize

The solution space must support a notion of step to move from one state to another. We operate on ranks; so the step unit is one, and the move can be accomplished in k steps. For each parameter, we have to calculate the number of steps relatively to Lévy flight value, using Mantegna algorithm in order to have stepsize between 0 and 1, we use a sigmoid function to transform the step size from real number space to probability space:

$$p(stepsize) = \frac{1}{(1+\exp(-stepsize))} \quad (18)$$

With adaptation p*(stepsize)* is exactly the step provided in Eq. 16. To facilitate control of these steps via the Lévy flight, we combine a range between 0 and 1 with step number. According to the value transformed in this range, we can choose the appropriate stride length.

For a parameter with n values; we define $d = \frac{1}{n}$, and k in $\{1, \ldots, n\}$. If the $p(stepsize)$ in $[(k-1) \times d, k \times d[$ then, the step number *is* k,

For example, if $n = 4$, then $d = 0.25$ and our interval is divided to four parts.

p(stepsize) in $[0, 1\times d[= [0, 0.25[$ then k=1.

p(stepsize) in $[1\times d, 2\times d[= [0.25, 0.50[$ then k=2.

p(stepsize) in $[2\times d, 3\times d[= [0.50, 0.75[$ then k=3.

p(stepsize) in $[3\times d, 4\times d[= [0.75, 1[$ then k=4.

For a parameter j, the rank update is:

$$Rank(u(t+1)_j) = modulo(Rank(u(t)_j) + k_j, |D_j|) + 1 \quad (19)$$

Then:

$$Rank(X(t+1)) = (Rank(u(t+1)_1), \ldots, Rank(u(t+1)_M) \quad (20)$$

We obtain the updated position using the inverse function:

$$X(t+1) = Value(Rank(X(t+1)) \quad (21)$$

4.2.6 The modified cuckoo search algorithm

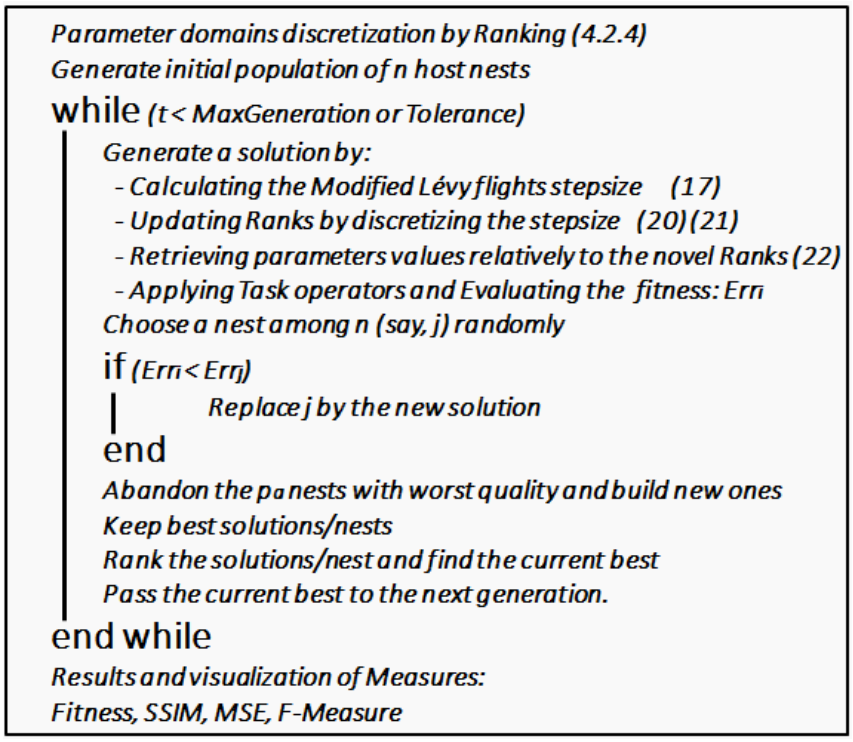


Figure 8. The modified ADCS Algorithm

Numbers between parentheses in Fig. 8 correspond to equations cited before.

5 Experiments

In the previous section the novel ADCS approach is described, the problem of choosing optimal operators and the optimal values of their parameters in a vision task is solved. The approach we presented in theory is applicable to any vision task that needs operators' selection or parameter adjustment or both of them.

Firstly, the experiment is conducted on a dataset of mechanical seals (e.g. car gasket). A seal consists of material and empty areas. To validate a manufactured seal, shapes and some measures must be inspected (positions, perimeters, surfaces, ...). A dedicated vision system must extract contours of empty areas. The second one consists of Berkeley database [31] of natural images. A processing task composed of several operators is proposed. To implement the system, the first step is to fix possible operators and possible values of their parameters. Then, the system runs in three phases (Fig. 9)

5.1 Phases, operators and parameters

The task of contour detection [32] allows identifying areas, of a digital image, corresponding to a brutal change in light intensity. It significantly reduces data quantity and eliminates the information judged less relevant, while preserving the important structural properties of the image, in order to extract measurements for example. This task is made of three phases of treatment (Fig. 9), firstly a pre-processing phase is necessary to remove any noise, then processing phase would take place to determine object contours, a post-processing phase will go after, to delete small and insignificant contours. The comparison uses ground truth images led by a CAD (Computer Aided Design) system for seals factory, and human segmentation for Berkeley database images.

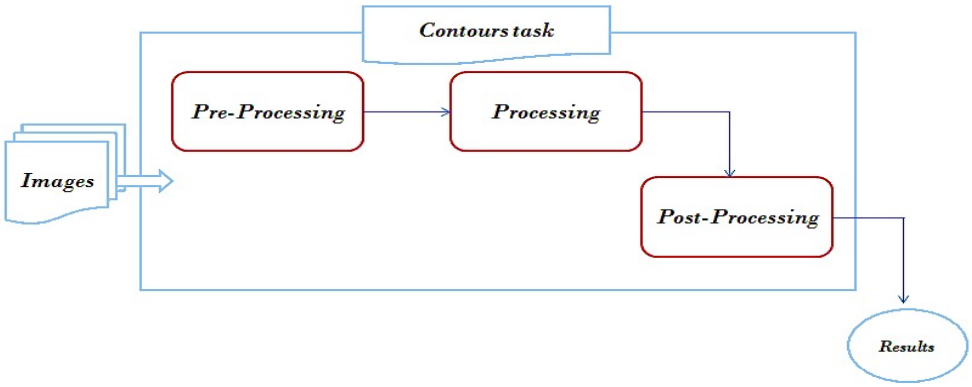


Figure 9. Phases of treatment of contours task

- **Pre-processing phase**

Pre-processing phase consists of improving image quality using filters; we propose a list of three filters predefined in Matlab: Meanshift and wiener2. Meanshift (named cvsm in our experiment) is a nonlinear filter, replaces each pixel with the mean of the pixels in a range-*r* neighborhood and whose value is within a distance *d*, this filter has two parameters to be adjusted. Wiener2 is an adaptive noise-removal filtering; it uses a pixel wise adaptive wiener method based on statistics estimated from a local neighbourhood of each pixel. The last two filters have just one parameter to be adjusted as indicated in Table. 1.

- **Processing phase**

Processing phase consists of detecting edges (contours), the operator applied is predefined in Matlab: edge with two parameters to be adjusted, filter to use and the threshold. A range of possible values would be provided for each parameter as in Fig. 10.

- **Post-processing phase**

Post-processing phase consists of refining the image by deleting small objects. The operator 'bwareaopen' would be applied as one operator of this phase; it is a morphological operator, which removes from a binary image all objects that have connectivity inferior than a predefined threshold. A range of possible thresholds and connectivity values would be provided as in Table. 1.

Number of phases : 3							
	Phase 1		Phase 2			Phase 3	
Operators	Meanshift	Wiener2	Sobel	Canny	Prewitt	Bwareaopen	
Possible parameter's values	0.5 1 2 3 4 5 6 7 8 9 11 15 16	3 5 7 9 11 13 15	0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 2 3 4 5 6 7	0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 2 3 4 5 6 7	0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 2 3 4 5 6 7	5 10 15 20 25 30 35 40 45 50	4 8

Table 1. Input possible values of defined operator's parameters.

Table 1 presents the input values for operators of this vision task and their parameters. Then six possible operators' combinations are generated:

C_1 = [Meanshift, Sobel, Bwareaopen]
C_2 = [Meanshift, Canny, Bwareaopen]
C_3 = [Meanshift, Prewitt, Bwareaopen]
C_4 = [Wiener2, Sobel, Bwareaopen]
C_5 = [Wiener2, Canny, Bwareaopen]
C_6 = [Wiener2, Prewitt, Bwareaopen]

Optimal parameters would be found for each one of these combinations according to the procedure described in Fig. 5, then the combination to be adopted is the one corresponding to the smallest error rate.

- **Objective function**

The objective function (error function) depends on the optimization problem. In this case, for the contour detection task, many adapted objective functions are possible. The error should indicate how good the input image segmentation is, in comparison to the reference image. In general, to compare two images, we use the MSE (Mean Square Error). It measures the average of errors squares that is the difference between the estimator and what is estimated, it is defined by:

$$MSE = \frac{1}{L*C}\sum_{i=1}^{L}\sum_{j=1}^{C}(O_{i,j} - D_{i,j})^2 \quad (22)$$

Where L represents rows' number and C is column number. In our case, *O* is the ground truth and *D* is the segmented image. Segmentation quality can also be measured by means to PSNR between the ground truth image and the segmented image. The value of PSNR is computed in dB using:

$$PSNR = 10Log_{10}\frac{(MaxIntensity)^2}{MSE} \quad (23)$$

Often, we use SSIM, Structural Similarity Index Measure, to compare segmented images to ground truth:

$$SSIM = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_{2x}+\mu_{2y}+C_1)(\sigma_{2x}+\sigma_{2y}+C_2)} \quad (24)$$

where μ_x, μ_y, σ_x, σ_y, and σ_{xy} are the local means, standard deviations, and cross-covariance for images x, y. C_1 and C_2 are constants.

Since the result images are only white and black pixels, we can compute the error function by using the confusion matrix for a two-class classifier. Several standard indicators have been defined for the two-class matrix; the one used in this work is the accuracy, which represents proportion of predictions total number that were correct. It is determined using the equation:

$$Acc = \frac{t_p+t_n}{t_p+t_n+f_p+f_n} \quad (25)$$

Where t_p (true positive) represents white pixels well classified (contours), t_n (true negative) represents dark pixels well classified, f_p (false positive) represents contours misclassified and f_n (false negative) represents dark pixels misclassified.

The F-measure is mainly used in the boundary-based evaluation [33]. Specifically, a precision-recall framework is introduced to compute this measure. Precision is the fraction of true positives detection rather than false positives, while recall is the fraction of true positives detected rather than missed.

Precision: $P = \frac{t_p}{t_p+f_p}$

Recall: $R = \frac{t_p}{t_p+f_n}$

$$\mathrm{F-measure} = \frac{P*R}{\alpha*R+(1-\alpha)*P} \qquad (26)$$

where α is a relative cost between precision (P) and recall (R). We set it to 0.5 in the experiment.

We are comparing, in this work, contour pixels found using ADCS approach to those in the ground truth image using two different measures for comparison. The error considered in this work is:

$$\boldsymbol{Err = 1- Acc} \qquad (27)$$

or

$$\boldsymbol{Err = 1-SSIM} \qquad (28)$$

We, also, observe **MSE** (for binary images, 1-***Acc*** is exactly ***MSE*)**, the **PSNR** and **F-measure** evolution. For each operators' combination *i*, applying DACS approach provides the best parameter values with the minimum error value.

$$\boldsymbol{Err^*(Ci)} = \min_{(p_j)} (ErrC_i(p_1, \ldots, p_m)) \qquad (29)$$

In the example above, we can enumerate six operators' combinations and the best combination realizes the minimum:

$$\min_i Err^*(C_i)$$

5.2 Results and Discussion

The experiment was conducted on a dataset of mechanical steal images, and natural images from Berkeley database. Parameters' values are defined and then the number of cuckoo's positions (combinations of parameter values) is about 1.166.400 !

The ADCS model is applied, and Fig. 10 reveal an image from Berkeley database with its reference done by experts in addition to ADCS approach result, in the same figure different plots of the objective function and other measures are illustrated, pointing error rate evolution for each possible operators' combination. The optimal one and the best parameters' values, is synonymous to the minimal error. Curves in Fig. 10 clearly show optimal operators' combination. Fig. 11 and Fig. 12 display random images used in this experience, and an error rate evolution plot of optimal operator's combination and their best parameters based on SSIM for both.

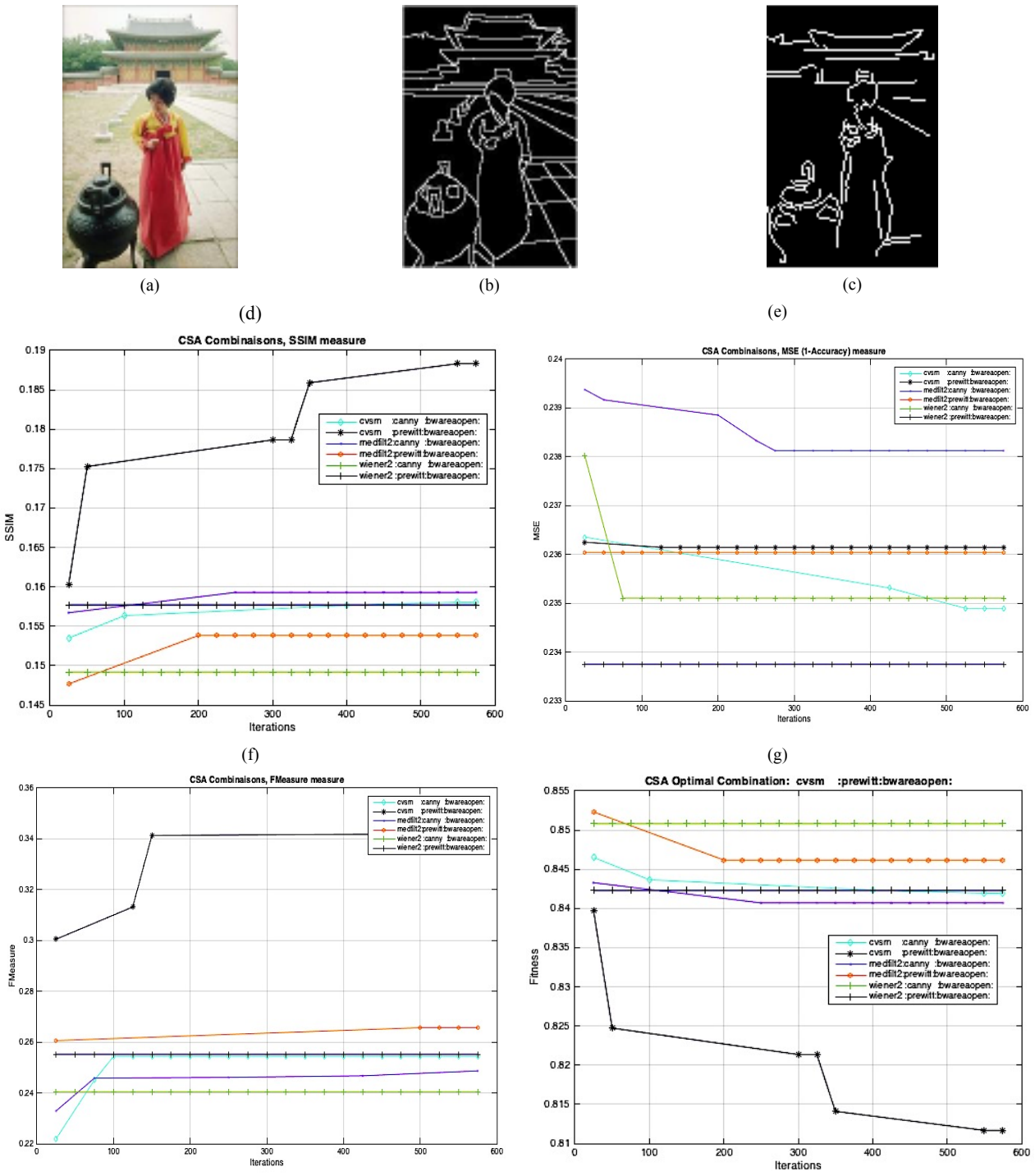
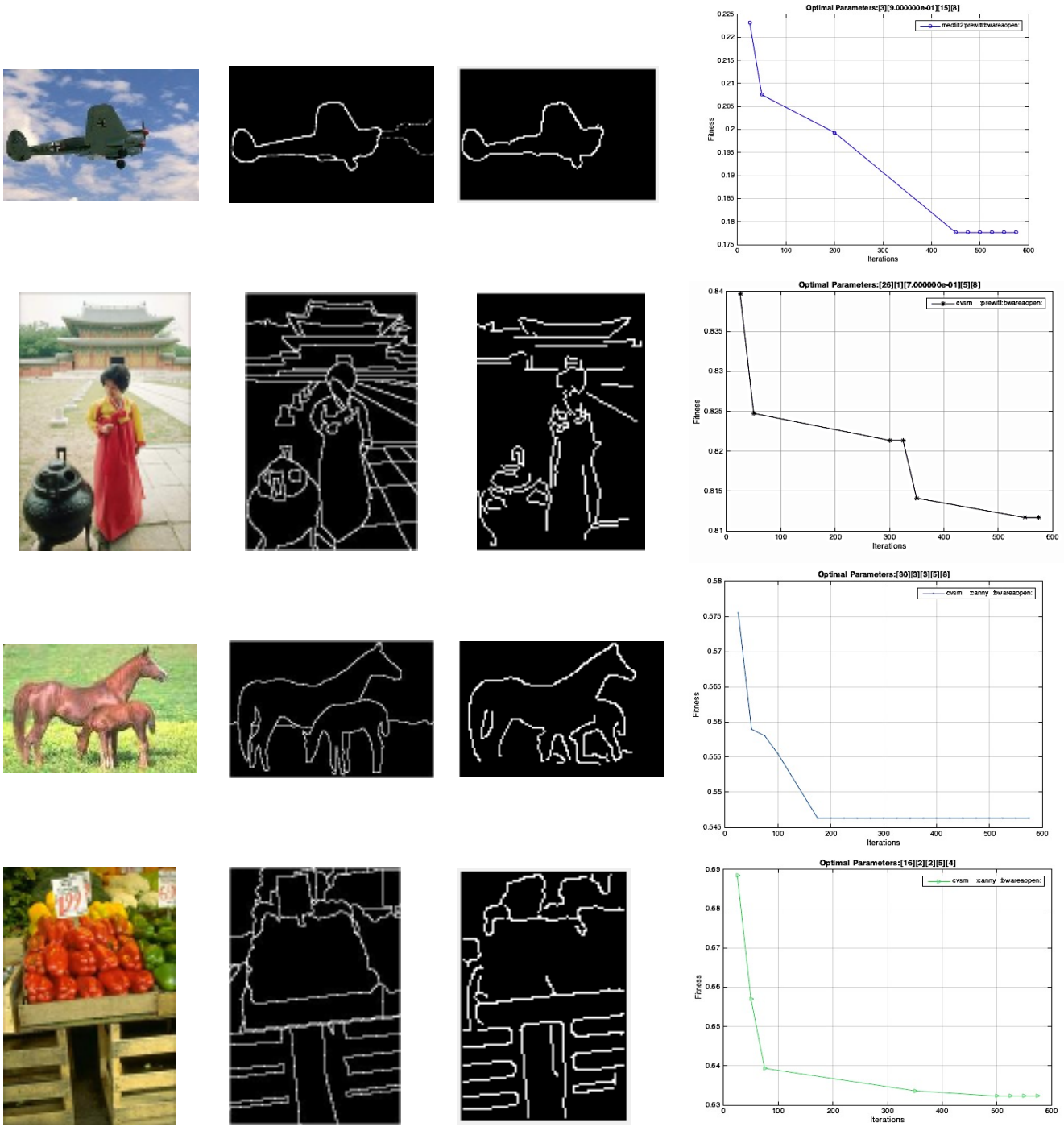
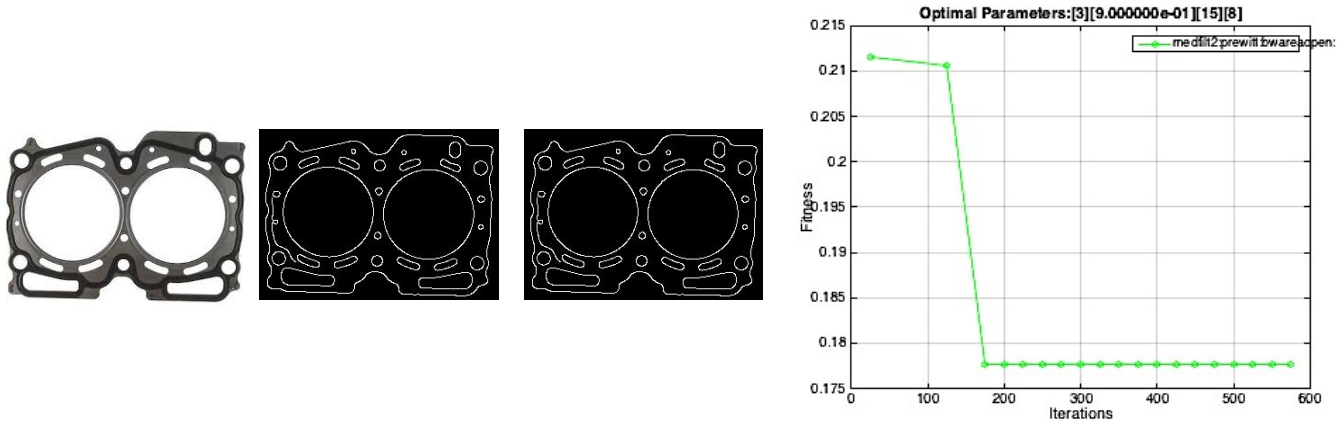


Figure 10. (a) An image from Berkeley database, (b) its reference done by experts, (c) ADCS approach result, (d) Evolution of SSIM measure for each operator's combination (e) Evolution of MSE (1-Accuracy) measure for each operator's combination, (f) Evolution of FMesure measure for each operator's combination, (g) Error evolution for each operator's combination based on SSIM measure.



a b c d

Figure 11. Images from Berkeley database: a) original image of Berkeley database b) Human segmentation image c) ADCS algorithm results d) Error evolution for best operators' combinations.



a b c d

Figure 12. Images of seals factory: a) original image of mechanical seals b) Reference image provided by the CAD system c) ADCS algorithm results d) Error evolution for the best operators' combinations.

The proposed method achieves good results with a small error rate. We note that the best results for Berkeley database are obtained with combinations using "Meanshift" operator. This indicates the importance of this operator in natural images' segmentation. However, for images of mechanical objects, the best results are obtained with practically the standard parameter values proposed by default.

5.3 ADCS parameters and comparison

Since tuning the parameters of an optimization algorithm is at least as important as the algorithm conception, it is therefore necessary to adjust iterations number and ADCS algorithm parameters. Some operators are time-consuming like "Meanshift", increasing the iteration number influences hardly the running time. Increasing the number of nests provides a good result in a small number of iterations, but increases running time (nests are consulted sequentially). Because of the stochastic aspect, the running time and convergence can vary from an execution to another (for the same image). But convergence can be stabilized by Beta value (β in Eq. 16) as shown in Fig.14, which plots the evolution of Error when varying beta values. The convergence is assured for $\beta <$ 2 (tests on the image (a) Fig.10).

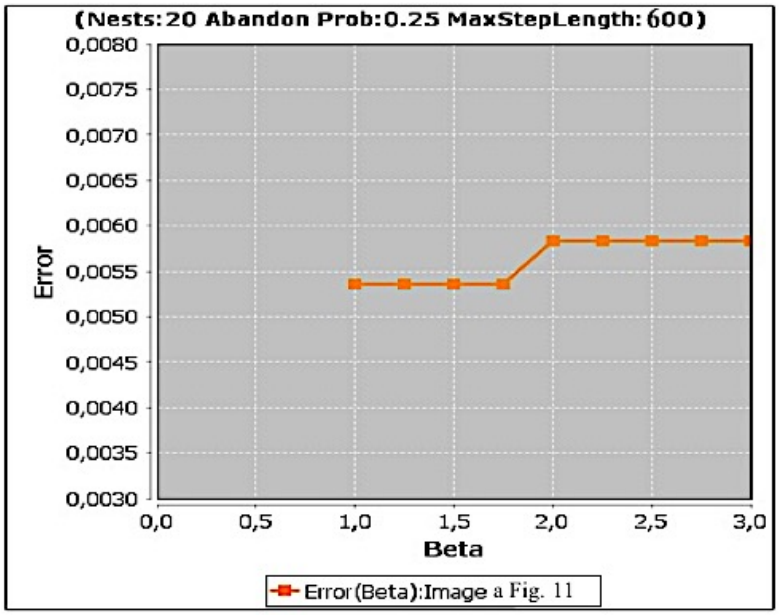


Figure 13. Error rate evolution with beta values image (a) in Fig 10.

To furthermore attest efficiency of ADCS approach, 100 images of Berkeley database and images of mechanical objects are used in this experience, keeping the same experimental setup described above. Results obtained using the approach proposed in this work, the novel Adaptive Discrete Cukoo Search Algorithm, are confronted to different approaches established in our previous works as discrete particle swarm optimization (DPSO) [34], ant colony optimization (ACO) [17], and another approach in literature such as reinforcement learning (RL) [6]. Fig. 14 shows the error rates obtained by the proposed ADCS approach in contrast to other techniques presented above, just 30 images are presented for the plot clarity. The error average for the 100 images using ADCS approach is $0{,}14x10^{-2}$ when the average error for DPSO approach is $0{,}17x10^{-2}$, for ACO approach the averange error of 100 images is $0{,}15x10^{-2}$ and for RL approach the average error is $0{,}18x10^{-2}$. The proposed ADCS obtains best results on the majority of test images, however other approach achieves good results also.

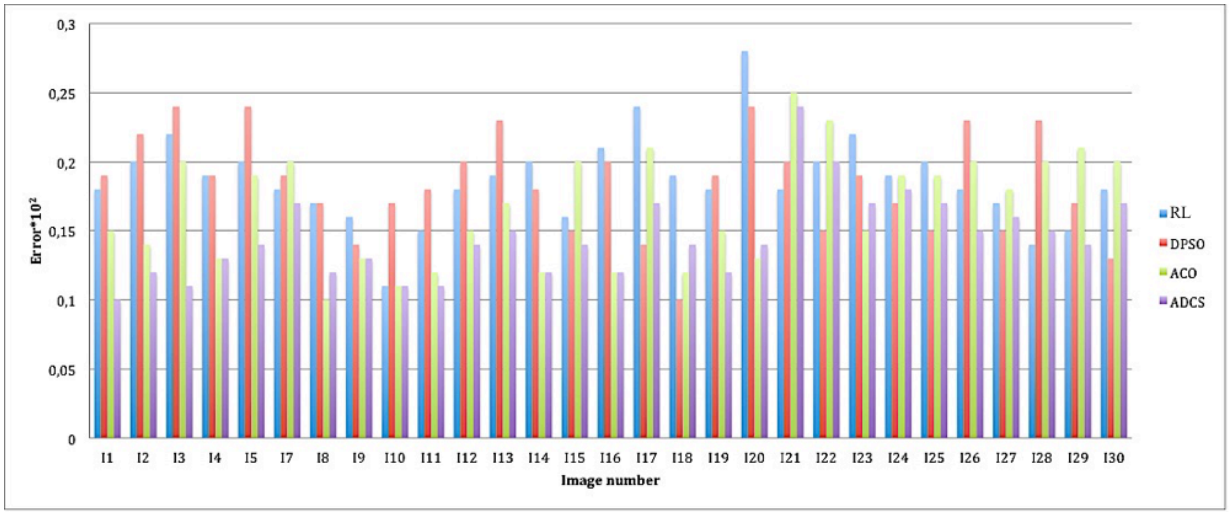


Figure 14. Bar plot of test error with different methods. ADCS stands for Adaptive Discrete Cuckoo Search algorithm proposed in this research, ACO denotes Ant Colony Optimization, RL means Reinforcement Learning and DPSO stands for Discrete Particle swarm optimization.

6 Conclusion

Choosing the appropriate operators to apply, and then adjusting their parameter values to accomplish a vision task is a very big challenge for users. In this work, we presented an automated method to optimize parameters' values of image processing algorithms. Our system proceeds automatically to decide which operator is the most appropriate to use, and optimizes automatically their free parameters. We presented a novel approach that adapts the CS algorithm to discrete problems, the novel approach is called adaptive discrete cuckoo search algorithm. In practice, tests showed the performance of such procedure. Our objective is intended to provide the best outcomes by tuning parameters via an open methodology as we proposed that can implement many adapted metaheuristics to discrete domains.

References

[1] Taylor GW. A Reinforcement Learning Framework for Parameter Control in Computer Vision Applications IEEE Proc. of the First Canadian Conf. on Computer and Robot Vision, 2004. Doi: 10.1109/CCCRV.2004.1301489

[2] Bao Y, Hu Z, Xiong T. A PSO and pattern search based memetic algorithm for SVMs parameters optimization Neurocomputing, pp 98–106, 2013. Doi: 10.1016/j.neucom.2013.01.027

[3] Benchikhi L, Sadgal M, El fazziki A. Optimization of parameters values in industrial vision algorithms Proc. of the International Symposium on Operational Research and Applications Marrakech, p 145-146, 2013.

[4] UEVAS E. Artificial Bee Colony (ABC) algorithm and its use in digital image processing. Inteligencia Artificial, [S.l.], v. 18, n. 55, p. 50-68, june 2015. ISSN 1988-3064.

[5] Benchikhi L, Sadgal M and Elfazziki A. An Optimization approach of parameters in image processing based on PSO: case of quality control Proc. Of the International conference on Industrial Engineering and Systems Management, Rabat, 2013.

[6] NÚÑEZ TABALE Julia M, REY CARMONA Francisco J, CARIDAD Y OCERIN José Mª. Artificial Intelligence (AI) techniques to analyze the determinants attributes in housing prices. Inteligencia Artificial, [S.l.], v. 19, n. 58, p. 23-38, dec. 2016. ISSN 1988-3064.

[7] Yang X.S, Deb S. Cuckoo Search via Lévy Flights Proceedings of World Congress on Nature & Biologically Inspired Computing, India, pp 210-214, 2009. Doi: 10.1155/2014/138760

[8] Yang S.X, Deb S. Engineering optimisation by Cuckoo Search International Journal of Mathematical Modelling and Numerical Optimisation. pp 330-343, 2010.

[9] Fister I, Yang X.S, Fister D. Cuckoo Search: A Brief Literature Review", Chapter Cuckoo Search and Firefly Algorithm: Theory and Applications, Studies in Computational Intelligence, vol. 516, pp. 49-62, 2014. Doi: 10.1007/978-3-319-02141-6_3

[10] Gherboudj A, Layeb A, and Chikhi S. Solving the knapsack problems by a discrete binary version of cuckoo search algorithm. International Journal of Bio-Inspired Computation, pp 229-236, 2012. Doi: 10.1504/IJBIC.2012.048063

[11] Tein L.H. Recent advancements of nurse scheduling models and a potential path, Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and its Applications, pp 395-409, 2010.

[12] Nickolay B, Schneider B, Jacob S. Parameter Optimization of an Image Processing System using Evolutionary Algorithms. Proc. of the 7th International Conf on Computer Analysis of Images and Patterns, Germany, 10–12, 1997. Doi: 10.1007/3-540-63460-6_173

[13] Treuillet S, Driouchi D and Ribereau P. Adjustment of parameters in an image processing chain by an experimental 2k-p factorial designs. Traitement du signal, 2004.

[14] Franek L, Jiang X. Orthogonal design of experiments for parameter learning in image segmentation Signal. Processing Signal Processing, Volume 93, Issue 6, Pages 1694–1704, 2013. Doi: 10.1016/j.sigpro.2012.08.016

[15] Glover F, Kochenberger G.A. Handbook of Metaheuristics, vol. 57 of International Series in Operations Research & Management Science. Kluwer, 2003.

[16] Montazeri-Gh, Morteza, Poursamad, and Ghalichi B. Application of genetic algorithm for optimization of control strategy in parallel hybrid electric vehicles. Journal of the Franklin Institute. Pp: 420-435, 2006. Doi: http://dx.doi.org/10.1016/j.jfranklin.2006.02.015

[17] Benchikhi L, Sadgal M, Elfazziki A and Mansouri F. An ant colony based model to optimize parameters in industrial vision. Electronic Letters on Computer Vision and Image Analysis, 16(1), 33-53. 2017 doi:https://doi.org/10.5565/rev/elcvia.957.

[18] Zhou Y and Huang Z. Artificial glowworm swarm optimization algorithm for TSP, Control and Decision, Vol. 27, No. 12, pp.1816–1821, 2013.

[19] Ouaarab A, Ahiod B and Yang X. Discrete version of the cuckoo search algorithm for the traveling salesman problem. Neural Comput & Applic, 2013. Doi: 10.1007/s00521-013-1402-2

[20] Dey N, Samanta S, Chakraborty S, Das A, Chaudhuri S. S, Suri J. S. Firefly algorithm for optimization of scaling factors during embedding of manifold medical information: an application in ophthalmology imaging. Journal of Medical Imaging and Health Informatics, 4(3), pp : 384-394, 2014. Doi: https://doi.org/10.1166/jmihi.2014.1265

[21] Balochian S, Ebrahimi E, Parameter Optimization via Cuckoo Optimization Algorithm of Fuzzy Controller for Liquid Level Control. Hindawi Publishing Corporation Journal of Engineering, 2013. Doi: http://dx.doi.org/10.1155/2013/982354

[22] Civicioglu P, Besdok B. A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms". Artif. Intell. Rev, Pp 315–346, 2013. Doi: 10.1007/s10462-011-9276-0

[23] Rajabioun R. Cuckoo Optimization Algorithm. In: Applied Soft Computing journal, vol. 11, pp. 5508 – 5518, 2011.

[24] Valian E, Tavakoli S, Mohanna S, Haghi A. Improved cuckoo search for reliability optimization problems. Comput. Ind. Eng, pp 459–468, 2013. Doi: http://dx.doi.org/10.1016/j.cie.2012.07.011

[25] Bektas T. The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega, 34(3), pp: 209-219, 2006. Doi: http://dx.doi.org/10.1016/j.omega.2004.10.004

[26] Mohammed J, Basalamah M, Majid A and Sid-Ahmed A. Computer Vision-Based Quality Inspection System of Transparent Gelatin Capsules in Pharmaceutical Applications, American Journal of Intelligent Systems, 2(1): 14-22, 2012. Doi: 10.5923/j.ajis.20120201.03
[27] Glover F, Kochenberger G.A. Handbook of Metaheuristics, vol. 57 of International Series in Operations Research & Management Science. Kluwer, 2003.
[28] Vasant P.M. Meta-Heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance, pp. 1-734, 2013.
[29] Mantegna R. Fast, Accurate algorithm for Numerical Simulation of Lévy Stable Stochastic Processes. Physical Review E, Vol. 49, No. 5, pp. 4677-4683, 1994. Doi: https://doi.org/10.1103/PhysRevE.49.4677
[30] Yongquan Z, Ouyang X, Jian X. A discrete cuckoo search algorithm for travelling salesman problem. Int. J. Collaborative Intelligence, Vol. 1, 2014. Doi: 10.1007/s00521-013-1402-F2
[31] Martin, D., Fowlkes, C., Tal, D., Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. International Conference on Computer Vision. pp. 416-423, 2001.
[32] Canny J. A computational approach to edge detection. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1986. Doi: 10.1109/TPAMI.1986.4767851
[33] Martin, D., Fowlkes, C., Malik, J. Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Trans. on Pattern Analysis and Machine Intelligence. vol. 26, pp. 530-549, 2004.
[34] Benchikhi L, Sadgal M, Elfazziki A, Mansouri F. A Discrete Particle Swarm Optimization to Estimate Parameters in Vision Tasks. International Journal of Advanced Computer Science and Applications, 2016. Doi: 10.14569/IJACSA.2016.070128