Dharwad
ज्ञानेन विकासः

CROW SEARCH ALGORITHM

Course code : DS352
Lecturer : Dr.Utkarsh khaire

<u>Members</u> :
Yashdeep Shetty (18BCS092) , 3rd year
Tej Aravind (18BCS089) , 3rd year

1. Abstraction :

Crow Search Algorithm is an innovative metaheuristic optimization algorithm. In this paper, chaotic maps are combined into Crow Search Algorithm to increase its global optimization. Ten variant chaotic maps are used and the sine chaotic map is found as the best choice for high dimensional problems. The novel Chaotic Crow Search Algorithm relies on the substitution of a random location of search space and the awareness parameter of crow with chaotic sequences. The results show that the chaotic maps are able to enhance the performance of the Crow Search Algorithm. Also the novel Chaotic Crow Search Algorithm outperforms the conventional Crow Search Algorithm, Genetic Algorithm, and Particle Swarm Optimization Algorithm from the point view of speed convergence and the function dimensions.

2. Introduction :

Nowadays, optimization can be considered as one of the most important and interesting research topics . It is inside the core processes of every aspect and can be found in almost all fields such as engineering, science, energy, computer, etc. Since the complexity of the real-world scientific and engineering problems increased, optimization becomes a big challenge in soft computing.we know that Metaheuristics Algorithm (MA) is very good at solving these NP problems and finding the optimal/near-optimal solution in real-time. These algorithms become very popular since their advantages are easy in implementation, avoiding local optima, and flexible and versatile. They can be considered as a black box, and can solve different problem types: single/multi-objective, constrained or unconstrained, and continuous/discrete. Generally speaking, MA can be categorized into two major classes: single-based / individual-based algorithms and population-based. Examples of single-based algorithms are Tabu Search (TS), Guided Local Search (GLS), and Pattern Search (PS) whereas Particle Swarm Optimization (PSO), Differential Search Algorithm (DSA) and Grey Wolf Optimizer (GWO) are examples of population-based algorithms.

Crow Search Algorithm (CSA) is a recent algorithm developed by Alireza Askarzadeh in 2016, which simulates the crow behavior in storing their food and retrieving it when they need it. Since its appearance, CSA has been widely used and applied to different optimization problems such as chemical engineering , medical, power energy, feature selection, and image processing .

Our paper represents all modifications that are made to CSA to not get struck at local optima and outperform CSA as well in terms of efficiency.

The main contributions of this paper can be listed below:

- A comprehensive review to CSA and CCSA has been done.
- All modifications to the original CSA have been highlighted.
- Number of challenges/ideas as a future work have been suggested.

3.Literature Survey :

These algorithms suffer from stacking in local optima and high computational time. EAs using their search agents can search the feature space adaptively to find the optimal solution. Some of these algorithms are gradient descent algorithm , discrete particle swarm optimization (DPSO) , tabu search, elephant herding optimization (EHO), firefly algorithm (FA), harmony search (HS), charged system search (CSS), bird swarm algorithm (BSA), animal migration optimization (AMO), teaching–learning based optimization (TLBO) , moth-flame optimization (MFO) , and gray wolf optimizer (GWO) . Although selecting the best feature subset has been significantly improved, there is still a need to motivate to push these presented work further. This work tries a novel hybrid approach, where chaos is embedded with CSA. The main contribution of this paper is that a chaotic binary version of CCSA is proposed to enhance the performance of CSA. In this hybrid approach, the chaotic search methodology is adopted to select the optimal feature subset which maximizes the classification accuracy and minimizes the feature subset length. Ten one-dimensional chaotic maps are adopted and replaced with random movement parameters of CSA. In this study, CCSA is used as a feature selection algorithm. The performance of the proposed approach is tested on 20 benchmark datasets. In addition, the performance of CCSA is compared with seven other meta-heuristic algorithms. The organization of this paper is as follows.we discussed abstraction in section 1. Introduction part about the algorithm in section 2. Motivation for algorithm in section 4. Experimental results and regarding discussions regarding the standard algorithm CSA and CCSA in section 5. Finally, a summary of our work in section 6 with respective references in section 7.

To the best of our knowledge, there are very few studies in literature that cover or list all CSA aspects, variants, and applications. Our research article aims to carry an understandable study for all CSA aspects, why scientists/researchers are motivated to use this algorithm to solve different real-world optimization problems .

4. Motivation :

The novel Chaotic Crow Search Algorithm relies on the substitution of a random location of search space and the awareness parameter of crow with chaotic sequences. The results show that the chaotic maps are able to enhance the performance of the Crow Search Algorithm. Feature selection is the important part for an algorithm in order to get best results and CCSA is applied to optimize the feature selection problem. Ten chaotic maps are employed during the optimization process of CSA. Experimental results reveal the capability of CCSA to find an optimal feature subset which maximizes the classification performance and minimizes the number of selected features. Moreover, the results show that CCSA is superior compared to CSA and the other algorithms. In addition, the experiments show that sine chaotic map is the appropriate map to significantly boost the performance of CSA.

5. Methodology :

5.1.Crow Search Algorithm CSA:

A new population-based algorithm called Crow Search Algorithm (CSA) was proposed by Askarzadeh, which simulates the hiding of food behavior of crow. Crow is an intelligent bird that can remember faces and warn its species in danger. One of the most evidence of their cleverness is hiding food and remembering its location. Moreover, the exploration and exploitation of CSA can be learned from Figure 1. Overall, the pseudocode of CSA can be modeled as shown in Algorithm 1, Figure 2 is the flowchart of CSA, and its main phases can be shown as follows:

1) Initializing initial positions of crows swarm randomly (**x**), their memory of what they think will be their position of the hidden food (**m**) .

2) A fitness function (objective function) is used to evaluate each crow's position of what they thought their food is hidden in (**m**) , Each crow stores its hiding place in its memory variable m_i .

3) Now we initialize some of our variables:

a) random (**r**) : $0 \le r_i \le 1$

b) flight (**f**) : fl (any random)

c) Awareness Probability (AP) : $0 \le AP_i \le 1$

4) Now you start an iteration and run these steps for T_{max}

5) For each iteration t_i

1. Every crow in the swarm starts to chase any other random crow j (because they are the best thieves) and updates its position.
2. Now the chased crow can understand if it's chased by AP and if he gets to know that it is being followed then it will deliberately wandering directions to fool the chasing crow or if it doesn't get to know then his hiding place that he thinks is the one will be compromised .
3. In each iteration then we update the values of **m** as per the fitness function like if the previous value of m was performing better or the currently updated position value (Depending on which one takes us more close to our objective) (For crows their hidden food or the ones they think was hidden for them)
4. Then we repeat steps a) to c) to find out best **m** value for every crow at last uptil T_{max} iteration.

5.2.Figures in CSA:

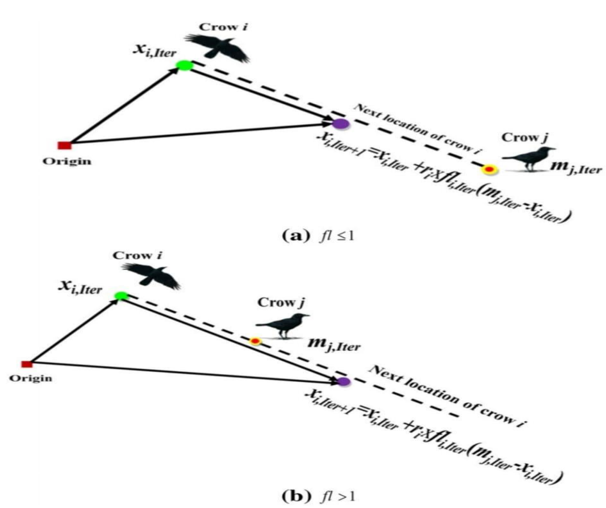


Fig 1. Exploration and exploitation mechanisms of CSA.

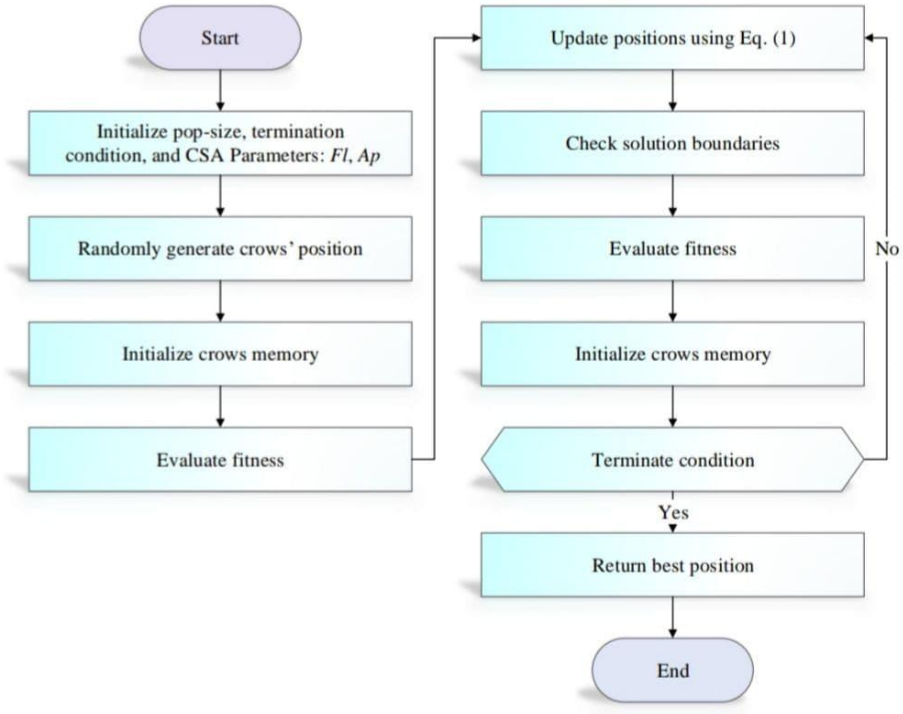


Fig 2. Flow chart for CSA algorithm.

5.3.Mathematical model of CSA:

In this section, we discuss the mathematical model of CSA and its research status/trend

$$x_{i,iter+1} = \begin{cases} x_{i,iter} + r_i \times \\ fl_{i,iter} \times \\ (m_{j,iter} - x_{i,iter}) & r_j \geq AP_{j,iter} \\ \\ \text{a random position} & otherwise \end{cases} \tag{1}$$

$$m_{i,iter+1} = \begin{cases} x_{i,iter+1} & f(x_{i,iter+1}) \leq f(m_{i,iter}) \\ \\ m_{i,iter} & otherwise \end{cases} \tag{2}$$

5.4.Algorithm of CSA:

Algorithm CSA :

Input n Number of crows in the population. $iter_{max}$ Maximum number of iterations.
Output Optimal crow position
Initialize position of crows.
Initialize crows' memory
while iter < $iter_{max}$ do
  for $crow_i$ belong to crows do
    choose a random crow.
    determine a value of awareness probability AP
    Update $x_{i,iter+1}$ using Eq.(1)
  end for
  Check solution boundaries.
  Calculate the fitness of each crow
  Update crows' memory using Eq.(2)
end while

5.5 Chaotic Crow Search Algorithm CCSA:

Chaos is defined as a phenomenon. Any change of its initial condition may lead to non-linear change for the future behavior. Chaos optimization is one of the recent search algorithms. The main idea of it is to transform parameters/variables from the chaos to the solution space. It depends on searching for global optimum on chaotic motion properties such as ergodicity, regularity, and stochastic properties. The main privileges of COA are fast convergence rate and their capability for avoiding local minima. All of these privileges can significantly improve the performance of evolutionary algorithms . Chaotic maps have a form of determinate, where no random factors are used .

Algorithm :

1) Initializing initial positions of crows swarm randomly (**x**), their memory of what they think will be their position of the hidden food (**m**) .

2) A fitness function (objective function) is used to evaluate each crow's position of what they thought their food is hidden in (**m**) , Each crow stores its hiding place in its memory variable m_i .

3) Now we initialize some of our variables:

a) random (**c**) : $0 \le c_i \le 1$

b) flight (**f**) : fl (any random)

c) Awareness Probability (AP) : $0 \le AP_i \le 1$

4) Now you start an iteration and run these steps for T_{max}

5) For each iteration t_i

I. Every crow in the swarm starts to chase any other random crow j (because they are the best thieves) and updates its position.

II. Now the chased crow can understand if it's chased by AP and if he gets to know that it is being followed then it will deliberately wander directions to fool the chasing crow or if it doesn't get to know then his hiding place that he thinks is the one will be compromised .

III. In each iteration then we update the values of **m** as per the fitness function like if the previous value of m was performing better or the currently updated position value (Depending on which one takes us more close to our objective) (For crows their hidden food or the ones they think was hidden for them)

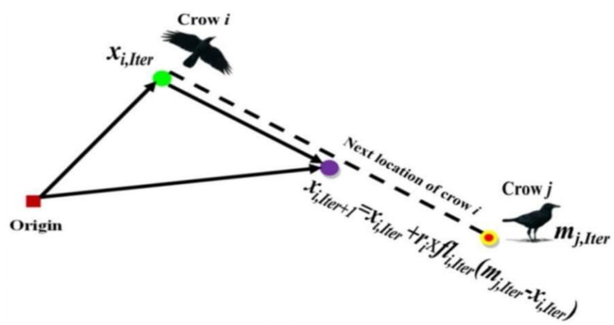
IV. Then we change the value of c as : $c_{t+1} = (c/4) * \sin(\pi Ct)$ where (small c=4)

V. Then we repeat steps I) to IV) to find out best **m** value for every crow at last Uptil $T_{max\ iteration}$.

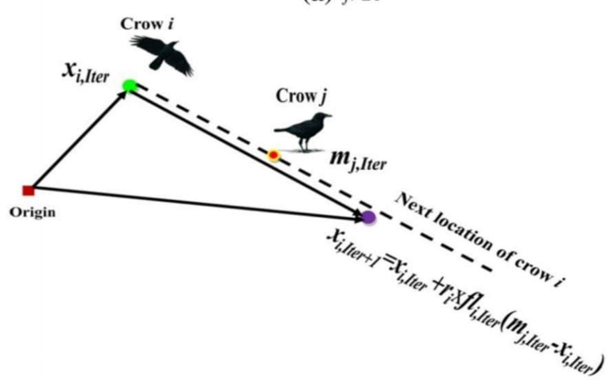
5.6.Figures in CCSA:

Table 1 The ten adapted chaotic maps

No.	Name	Definition	Range
CCSA1	Chebyshev	$p_{q+1} = \cos(q \cos^{-1}(p_q))$	(−1,1)
CCSA2	Circle	$p_{q+1} = \bmod\left(p_q + d - \left(\frac{c}{2\pi}\right)\sin(2\pi p_q), 1\right), c = 0.5 \text{ and } d = 0.2$	(0,1)
CCSA3	Guass/mouse	$p_{q+1} = \begin{cases} 1, & p_q = 0 \\ \frac{1}{\bmod(p_q, 1)}, & \text{otherwise} \end{cases}$	(0,1)
CCSA4	Iterative	$p_{q+1} = \sin\left(\frac{c\pi}{p_q}\right), c = 0.7$	(−1,1)
CCSA5	Logistic	$p_{q+1} = cp_q(1 - p_q), \mathrm{c} = 4$	(0,1)
CCSA6	Piecewise	$p_{q+1} = \begin{cases} \frac{p_q}{l}, & 0 \leq p_q < l \\ \frac{p_q - l}{0.5 - l}, & l \leq p_q < 0.5 \\ \frac{1 - l - p_q}{0.5 - l}, & 0.5 \leq p_q < 1 - l \\ \frac{1 - p_q}{l}, & 1 - l \leq p_q < 1 \end{cases}, l = 0.4$	(0,1)
CCSA7	Sine	$p_{q+1} = \frac{c}{4}\sin(\pi p_q), c = 4$	(0,1)
CCSA8	Singer	$p_{q+1} = \mu(7.86 p_q - 23.31 p_q^2 + 28.75 p_q^3 - 13.302875 p_q^4), \mu = 1.07$	(0,1)
CCSA9	Sinusoidal	$p_{q+1} = c p_q^2 \sin(\pi p_q), c = 2.3$	(0,1)
CCSA10	Tent	$p_{q+1} = \begin{cases} \frac{p_q}{0.7}, & p_q < 0.7 \\ \frac{10}{3}(1 - p_q), & p_q \geq 0.7 \end{cases}$	(0,1)



(a) $fl \leq 1$



(b) $fl > 1$

5.7.Mathematical Model of CCSA:

In this section, we discuss the mathematical model of CCSA and its research status/trend .

$$y^{j,t+1} = \begin{cases} y^{j,t} + C_j \times fl^{j,t} \times (N^{z,t} - y^{j,t}), & C_z \geq AP^{j,t} \\ \text{Choose a rand position ,} & \text{otherwise} \end{cases} \quad (1)$$

$$m_{i,iter+1} = \begin{cases} x_{i,iter+1} & f(x_{i,iter+1}) \leq f(m_{i,iter}) \\ m_{i,iter} & otherwise \end{cases} \quad (2)$$

5.8.Algorithm of CCSA:

Algorithm CCSA :

Set the initial values of M, AP, fl, and $tMax$.
Initialize the crow position y randomly.
Evaluate the fitness function of each crow $Fn(y)$.
Initialize the memory of search crow N
Set $t := 1$. {Counter initialization}.
repeat
  for ($j = 1 : j \leq M$) **do**
    Get value of chaotic map C
    if $C_z \geq AP^{z,t}$ **then**
      $y^{j,t+1} = y^{j,t} + C_j \times fl^{j,t} \times (N^{z,t} - y^{j,t})$
    else
      $y^{j,t+1}$ $=$*A random position of the search*
space

---

  end for
  Check the feasibility of $y^{j,t+1}$
  Evaluate the new position of crow $Fn(y^{j,t+1})$
  Update the crow's memory $N^{j,t+1}$
  Set $t = t + 1$. {Iteration counter increasing}.
until ($t < tMax$). {Termination criteria satisfied}.
Produce the best solution N.

---

6. Application (CCSA) :

A) Problem Statement : We took an objective function 30x-x^2 (Maximize) and applied our CCSA with search space(0-40) :

B) Coded (CCSA) :

```
#1. Set the initial values of M , fl and ap and tMax

M = 3
ap = [0.1 for i in range(M)]
fl = [2 for i in range(M)]
tMax = 100

# 2. Initialize the crow position (y) randomly , position of
hiding place in memory (N)  randomly along with C for each crow

yt = [5,25,11] #search space [0-40]
Ct = [0.2,0.2,0.5]
Nt = [10,18,14]
ytplus1 = []
Ntplus1=[]
Ctplus1=[]
Fn=[]

N=[]
# 3. Evaluate the fitness function of each crow Fn(y).
for i in range(M):
    Fn.append(objectiveFunction(yt[i]))

# 4. Set t = 1 and run till tMax .
t = 1
while t<=tMax:
    for j in range(M):
        # Randomly choose one of crows to follow
```

```
                z = random.randint(0, M-1)
                if (Ct[z] > ap[z]):
                    ytplus1.append(yt[j] + Ct[j] * fl[j] * (Nt[z] -
yt[j]))
                else:
                    ytplus1.append(random.randint(0, 40))
            # Check Feasibility
            for i in range(M):
                if(ytplus1[i]>=0 and ytplus1[i]<=40):
                    pass
                else:
                    ytplus1[i]=yt[i]
            # Updating Each Crow's Memory
            for i in range(M):

if(objectiveFunction(ytplus1[i])<objectiveFunction(Nt[i])):
                    Ntplus1.append(Nt[i])
                else:
                    Ntplus1.append(ytplus1[i])
            # Evaluate new position of crow n
            for i in range(M):
                Ctplus1.append(np.sin(np.pi*Ct[i]))

            Ct = Ctplus1
            yt = ytplus1
            Nt = Ntplus1
            Ctplus1 = []
            ytplus1 = []
            Ntplus1 = []
            print("N after iteration no.",t," ",Nt)
            N.append(Nt)
            t=t+1
        print("Final optimal N for each crow :")
        print(Nt)
```

C)Figure for Result(CCSA) :

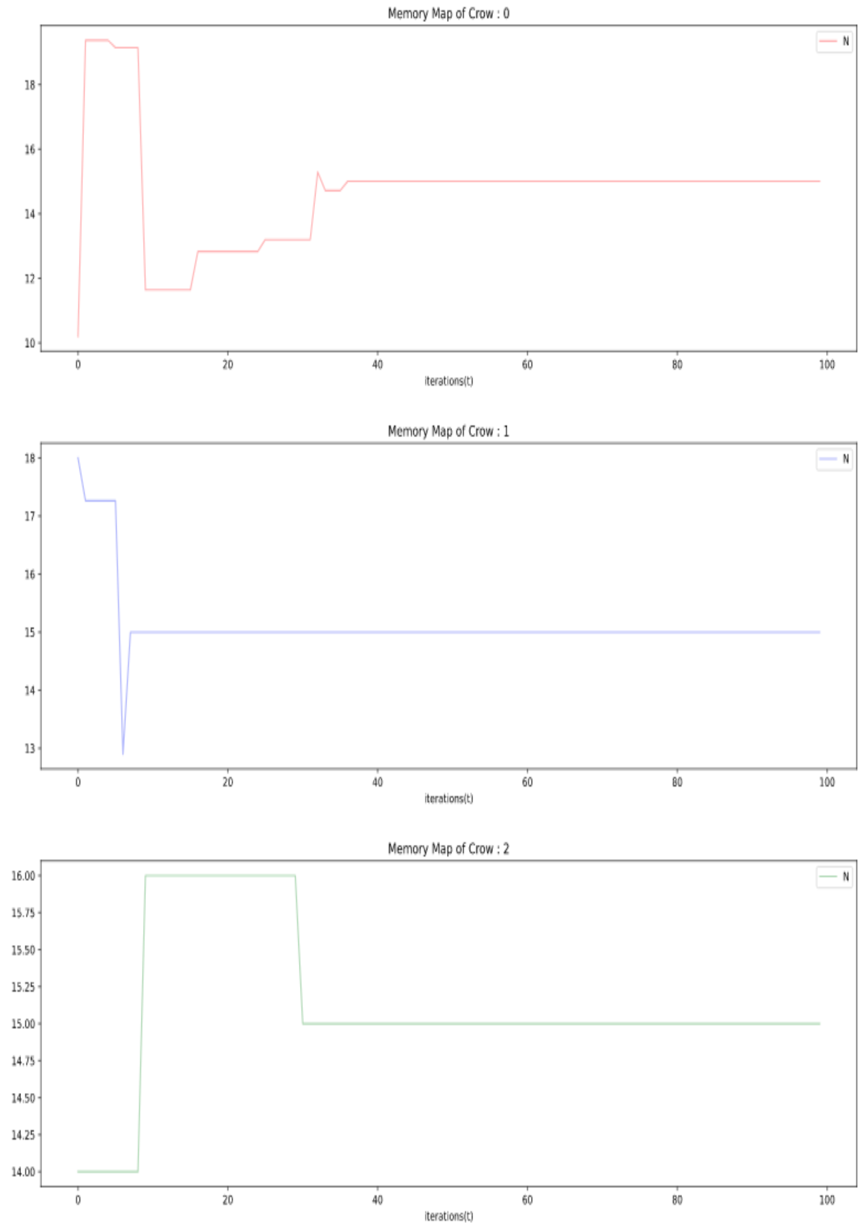


Fig - It shows the improvement of hiding Place Location compared to iteration for our 3 crows (i.e. close to 15 - our optimal value for objective function)

7. Conclusion :

In this paper, a novel hybridization of chaos with the CSA algorithm, namely CCSA, is proposed. We took an objective function 30x-x^2 (Maximize) and in the attempt to find an optimal value of x , for each crow the values of x were closed to 15 the optimal point where the global maxima of this function lies. Further work on embedding chaotic maps with other meta-heuristic algorithms will be considered. For future verification, the performance of CCSA will be applied on more complex science and real-world engineering problems. Moreover, other chaotic maps are also worth applying to CSA.

8. References :

1. https://drive.google.com/file/d/1MYYmpzrFUFi9LrW2mBZDh8hP4ZhewfYz/view for CCSA.
2. https://www.slideshare.net/afar1111/crow-search-algorithm for comprehensive CSA .