الگوریتم کلونی زنبور عسل مصنوعی
(ABC)

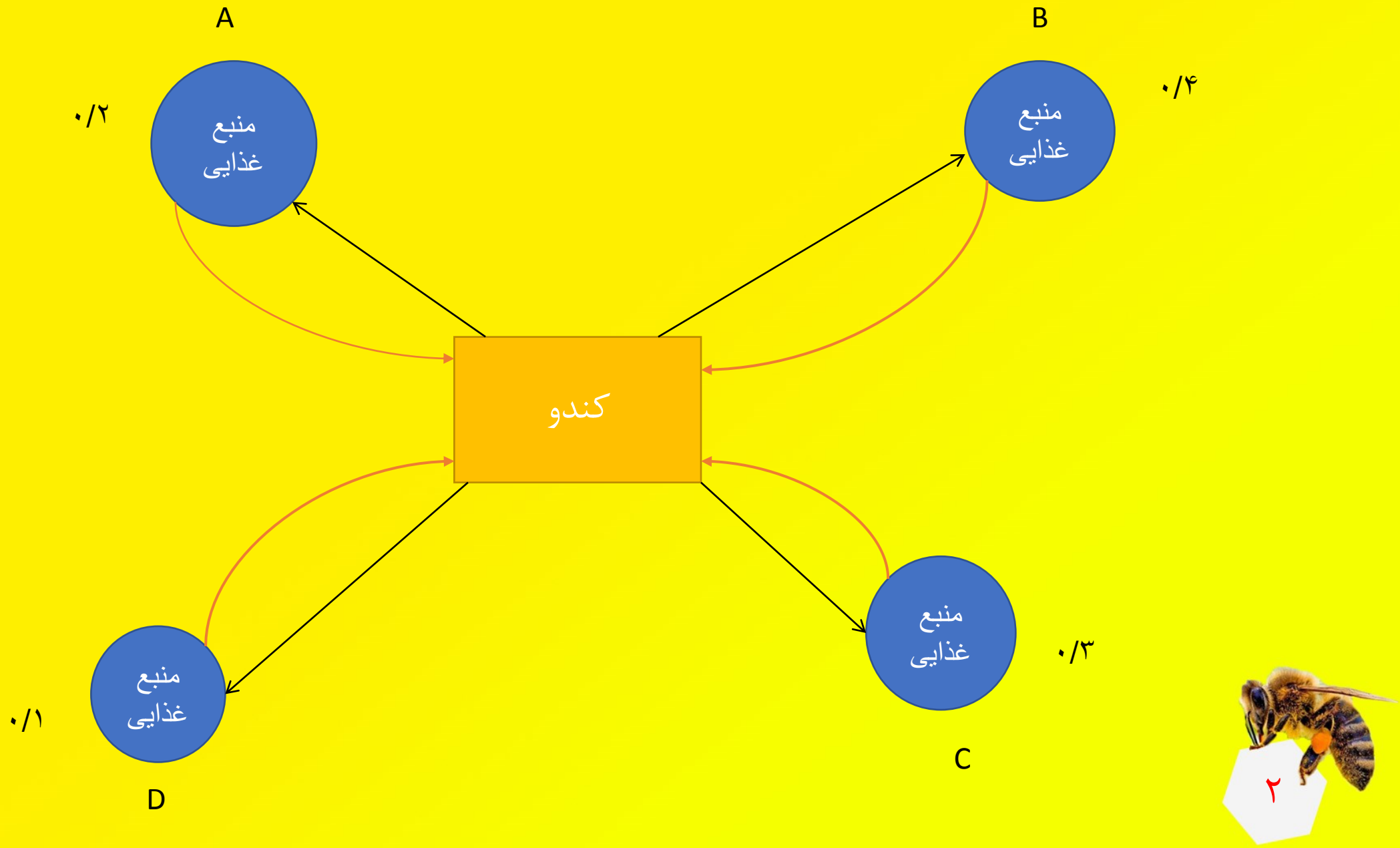
Artificial Bee Colony

بسم الله الرحمن الرحیم

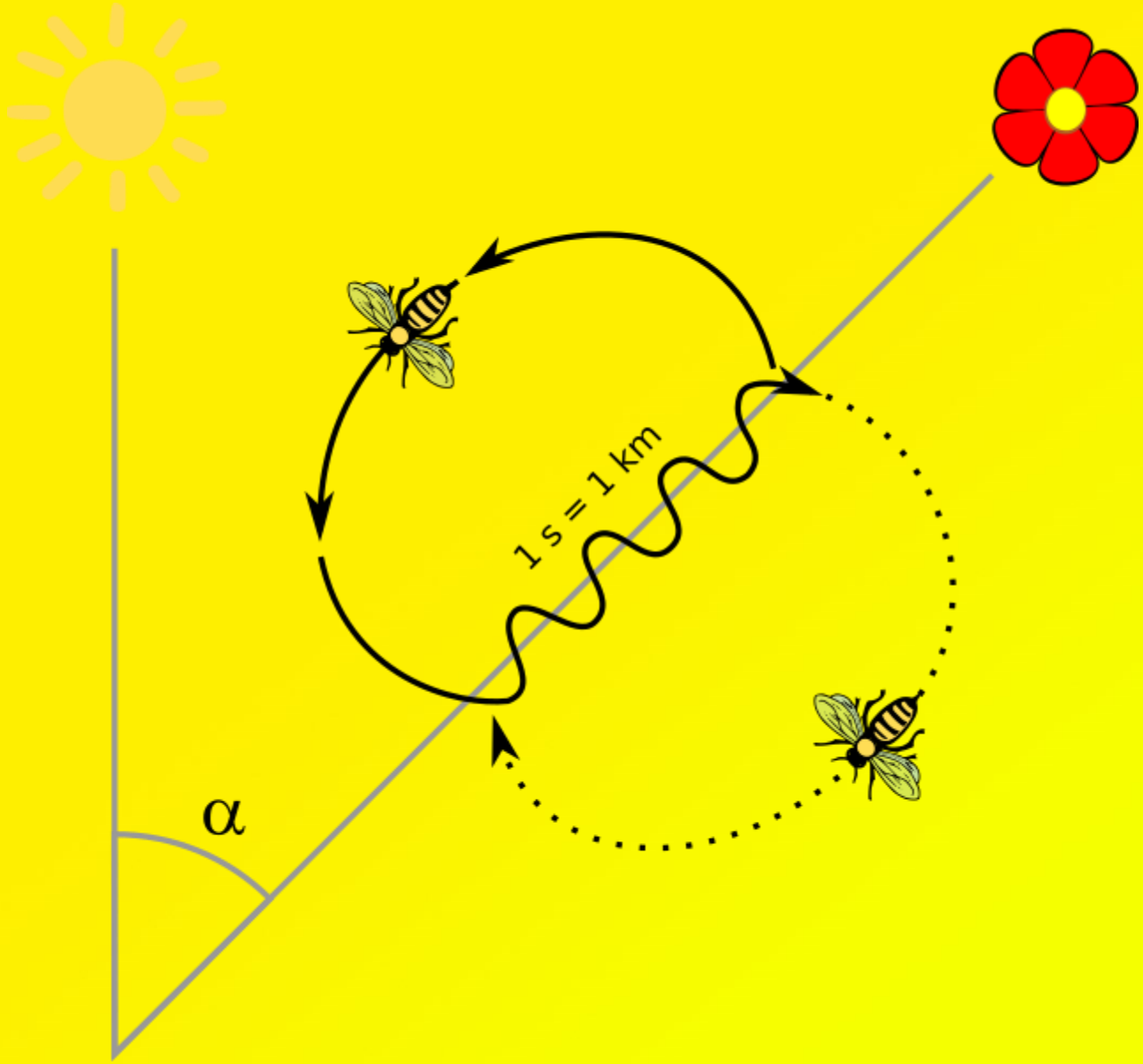
الگوریتم کلونی زنبور عسل مصنوعی (ABC)، یک راهکار بهینه‌سازی است که رفتار یک کلونی زنبور عسل را شبیه‌سازی می‌کند و برای اولین بار در سال ۲۰۰۵ توسط «کارابوگا» (Karaboga)، برای بهینه‌سازی پارامتر واقعی ارائه شد.

در این مدل ریاضی سه نوع زنبور وجود دارد:

۱. زنبورهای کارگر(employed bees)
۲. زنبورهای ناظر(onlooker bees)
۳. زنبورهای پیشاهنگ(scout bees)


A
B
۰/۲
۰/۴
منبع غذایی
منبع غذایی
کندو
منبع غذایی
منبع غذایی
۰/۳
۰/۱
C
D
۲

رقص زنبورعسل



مراحل الگوریتم کلونی زنبورعسل:

۱) تولید پاسخ‌های تصادفی اولیه و ارزیابی آن‌ها

۲) حرکت زنبورهای استخدام‌شده

$$j \in \{1, 2, \dots, D\}$$

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \qquad k \neq i$$

$$k \in \{1, 2, \dots, n_{pop}\}$$

$$\phi_{ij} \sim U(-a, +a)$$

$$v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$$

$$x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$$

$$x_i \leftarrow \begin{cases} v_i & \text{اگر } v_i \text{ بهتر از } x_i \text{ باشد} \\ x_i & \text{اگر } x_i \text{ بهتر از } v_i \text{ باشد} \rightarrow C_i \text{++} \end{cases}$$

۳) ارسال زنبورهای ناظر

$f(x_i)$ ⟶ مقدار تابع هزینه پاسخ x_i

$$F(x_i) = \begin{cases} \frac{1}{1+f(x_i)} & f(x_i) \geq 0 \\ 1+|f(x_i)| & f(x_i) < 0 \end{cases}$$

مقدار برازندگی

(Artificial Bee Colony) در متلب

$$P_i \propto F(x_i) \geq 0$$

$$P_i = \frac{F(x_i)}{\sum_{k=1}^{n} F(x_k)}$$

$$\begin{cases} 0 \leq P_i \leq 1 \\ \sum_{i=1}^{n} P_i = 1 \end{cases}$$

محاسبه احتمالات ← انتخاب سایت ← تولید پاسخ جدید ← مقایسه

به تعداد اعضای جمعیت

۴) اگر سایتی وجود دارد که مقدار دفعات عدم پیشرفت آن به L رسیده باشد، آن سایت را با یک پاسخ تصادفی جایگزین می کنیم و شمارنده مربوط به آن را برابر با صفر قرار می دهیم

۵) در صورتی که شرایط خاتمه برآورده نشده‌اند، به مرحله ۲ برمی‌گردیم. درغیر این صورت پایان.

به دست آوردن کمینه‌ی تابع sphere:

Sphere $f_{sph}(x) = \sum_{i=1}^{n} x_i^2$

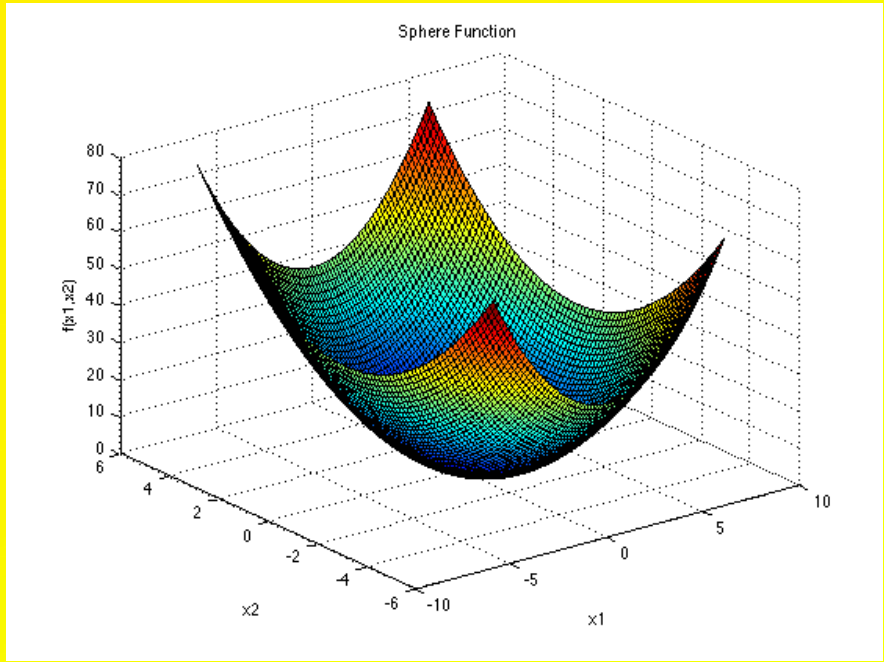
Cost Function

$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$

n_{var}

$x_{min}^i \leq x_i \leq x_{max}^i$

VarMin VarMax



$x_1(2,2)$ $x_2(5,1)$ $x_3(5,4)$

$f(x)=sphere(x)$ $f(x)=5^2+1=10$ $f(x)=5^2+4^2=25+16=41$

$f(x)=2^2+2^2=12$

Bestsol $=x_2$ Best cost $=10$

$v_{ij}=x_{ij}+\phi(x_{kj}-x_{ij})$

$v_{11}=2+\frac{1}{2}(5-2)=2+1.5=3.5$ $v_{12}=4+\frac{1}{3}(9-2)=4$

$v_1=(3.5,4)$ $f(x)=(3.5)^2+4^2=28.25$ $c_1=1$

$v_{21}=3+\frac{1}{2}(2-5)=3-\frac{1}{2}=2.5$ $v_{22}=1+\frac{1}{2}(3-1)=2$

$v_2(2.5,2)$ $f(x)=(2.5)^2+2^2=10.25$ $c_2=1$

$v_{31}=5+\frac{1}{3}(3-5)=4$ $v_{32}=9+\frac{1}{3}(3-4)=5$

$v_3(4,5)$ $f(x)=4^2+5^2=41$ $c_3=0$

$x_1(2,2)$ $x_2(5,1)$ $x_3(4,5)$

$F(x)=\frac{1}{1+12}=\frac{1}{13}$ $F(x)=\frac{1}{1+10}=\frac{1}{11}$ $F(x)=\frac{1}{1+41}=\frac{1}{42}$

$P_1=\frac{\frac{1}{13}}{\frac{1}{13}+\frac{1}{11}+\frac{1}{42}}=\frac{7}{31.8}=.48$ $P_2=\frac{1}{2}=.5$ $P_3=\frac{1}{9}=.11$

2022 / ۱۴۰۰

$v'_{11} = 2 + \frac{1}{2}(3-2) = 2.5$ $\quad v'_{12} = 3 + \frac{1}{2}(1-3) = 2$

$v'_1 = (2.5, 2)$ $\quad f(x) = (2.5)^2 + 2^2 = 10.25$ $\quad x_1(2.5, 2)$

$c_1 = 1$

$v''_{11} = 2.5 + \frac{1}{2}(4-2.5) = 2.5 + 0.5 = 3$

$v''_{12} = 2 + \frac{1}{2}(5-2) = 4$ $\quad v''_2(3, 4)$ $\quad f(x) = 18$ $\quad x_1(2.5, 2)$

$c_1 = 2$

$v'_{21} = 3 + \frac{1}{2}(2.5-3) = 3 + \frac{1}{4} = 3.25$

$v'_{22} = 1 + \frac{1}{4}(2-1) = 1 + \frac{1}{4} = 1.25$

$v'_2(3.25, 1.25)$ $\quad f(x) = (3.25)^2 + (1.25)^2 = 10.56 + 1.56 = 11.92$

$c_2 = 2$ $\quad x_2(3, 1)$

جواب‌های این مرحله $\quad x_1(2.5, 2)$ $\quad x_2(3, 1)$ $\quad x_3(4, 5)$

Best sol $= x_2$ $\quad$ Best cost $= 10$

```
function z=Sphere(x)

    z=sum(x.^2);

end
```

```
clc;
clear;
close all;

%% Problem Definition

CostFunction=@(x) Sphere(x);        % Cost Function

nVar=5;                 % Number of Decision Variables

VarSize=[1 nVar];   % Decision Variables Matrix Size

VarMin=-10;          % Decision Variables Lower Bound
VarMax= 10;          % Decision Variables Upper Bound
```

```
%% ABC Settings

MaxIt=100;                  % Maximum Number of Iterations

nPop=10;                   % Population Size (Colony Size)

nOnlooker=nPop;             % Number of Onlooker Bees

L=round(0.5*nVar*nPop); % Abandonment Limit Parameter

a=1;                        % Acceleration Coefficient Upper Bound
```

```
%% Initialization

% Empty Bee Structure
empty_bee.Position=[];
empty_bee.Cost=[];

% Initialize Population Array
pop=repmat(empty_bee,nPop,1);

% Initialize Best Solution Ever Found
BestSol.Cost=inf;

% Create Initial Population
for i=1:nPop

pop(i).Position=unifrnd(VarMin,VarMax,VarSize
);

pop(i).Cost=CostFunction(pop(i).Position);

    if pop(i).Cost<=BestSol.Cost
            BestSol=pop(i);
        end
    end

    % Abandonment Counter
    C=zeros(nPop,1);

    % Array to Hold Best Cost Values
    BestCost=zeros(MaxIt,1);
```

```
%% ABC Main Loop

for it=1:MaxIt

    % Recruited Bees
    for i=1:nPop

        % Choose k randomly, not equal to i
        K=[1:i-1 i+1:nPop];
        k=K(randi([1 numel(K)]));

        % Define Acceleration Coeff.
        phi=unifrnd(-a,+a,VarSize);

        % New Bee Position
        newbee.Position=pop(i).Position+phi.*(pop(i).Position-pop(k).Position);
```

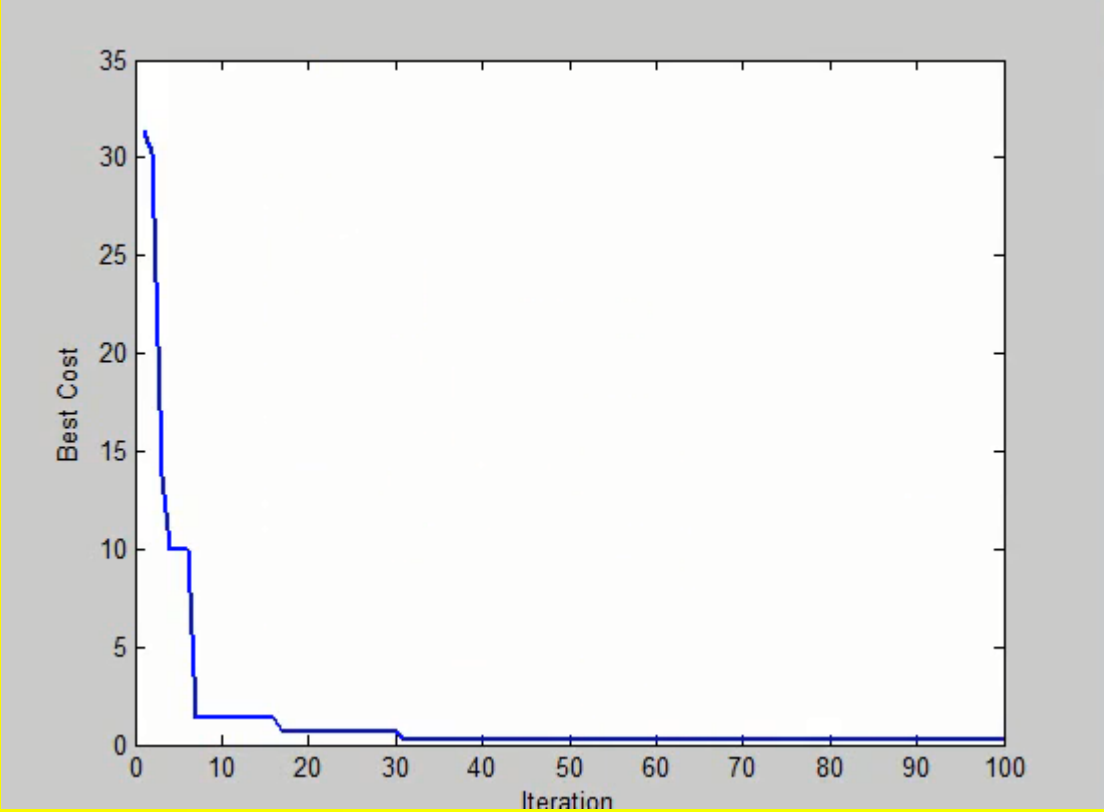
```
% Evaluation
        newbee.Cost=CostFunction(newbee.Position);

        % Comparision
        if newbee.Cost<=pop(i).Cost
            pop(i)=newbee;
        else
            C(i)=C(i)+1;
        end

    end

    % Calculate Fitness Values and Selection Probabilities
    F=zeros(nPop,1);
    for i=1:nPop
        if pop(i).Cost>=0
            F(i)=1/(1+pop(i).Cost);
        else
            F(i)=1+abs(pop(i).Cost);
        end
    end
    P=F/sum(F);
```

```
% Onlooker Bees
    for m=1:nOnlooker

        % Select Source Site
        i=RouletteWheelSelection(P);

        % Choose k randomly, not equal to i
        K=[1:i-1 i+1:nPop];
        k=K(randi([1 numel(K)]));

        % Define Acceleration Coeff.
        phi=unifrnd(-a,+a,VarSize);

        % New Bee Position
        newbee.Position=pop(i).Position+phi.*(pop(i).Position-pop(k).Position);

        % Evaluation
        newbee.Cost=CostFunction(newbee.Position);

        % Comparision
        if newbee.Cost<=pop(i).Cost
            pop(i)=newbee;
        else
            C(i)=C(i)+1;
        end
    end
```

```
function i=RouletteWheelSelection(P)

    r=rand;

    C=cumsum(P);

    i=find(r<=C,1,'first');

end
```

```
% Scout Bees
    for i=1:nPop
        if C(i)>=L
            pop(i).Position=unifrnd(VarMin,VarMax,VarSize);
            pop(i).Cost=CostFunction(pop(i).Position);
            C(i)=0;
        end
    end

    % Update Best Solution Ever Found
    for i=1:nPop
        if pop(i).Cost<=BestSol.Cost
            BestSol=pop(i);
        end
    end
```

```
    % Store Best Cost Ever Found
    BestCost(it)=BestSol.Cost;

    % Display Iteration Information
    disp(['Iteration ' num2str(it) ': Best Cost = ' num2str(BestCost(it))]);
end

%% Results

figure;
%plot(BestCost,'LineWidth',2);
semilogy(BestCost,'LineWidth',2);
xlabel('Iteration');
ylabel('Best Cost');
```



منابع:

1- https://virgool.io
2- https://blog.faradars.org
3- https://behsanandish.com
4- https://faradars.com
5- https://persianv.com
6- https://atiyesalamat.com
7- https://sid.ir
8- https://www.sciencedirect.com

۱-ارزیابی کارایی الگوریتم کلونی زنبورمصنوعی در حل مسائل بهینه سازی ترکیبی(امیرمسعود رحیمی*(استادیار)وفرشاد حمیدی(کارشناسی ارشد)/گروه عمران،دانشکده مهندسی،دانشگاه زنجان)