Minimize $f(X) = x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$

where $-5 \leq x_1, x_2 \leq 5$

The movement of the bees is recorded in three phases

Employed Phase-

- Generate a new solution.
- Calculate new fitness.
- Apply greedy selection.

Onlooker Phase-

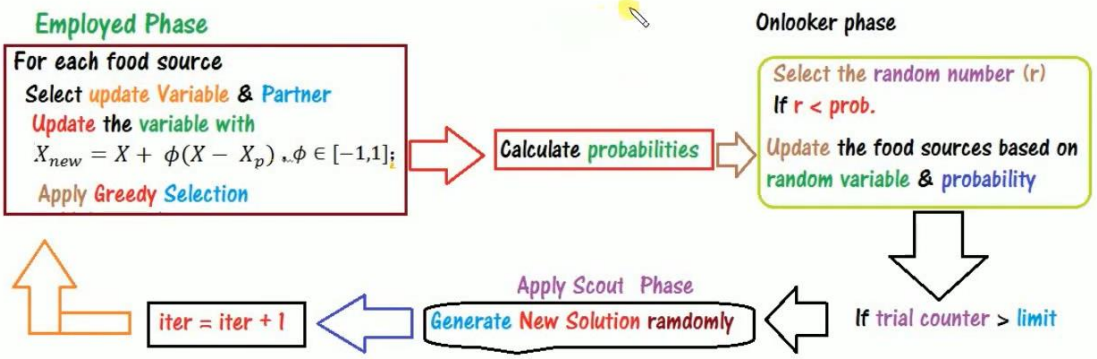
- Calculate the probabilities.
- Produce a new solution depending on probability.
- Calculate new fitness.
- Apply greedy selection.

Scout Phase-

Find the abandoned solution (based on the value of *limit*)

- Generate a new solution randomly to replace them.

Working Rule (Algorithm)



Phases of the MATLAB Code:

Phase 1: Input parameter

Phase 2: Defining Objective & Fitness Function

Phase 3: Generate Initial Population

ABC Main ITERATION Loop Start

Phase 4: Perform Employed & Onlooker Phases

- Select variable & Partner
- Generate New Solution
- Perform Greedy Selection

Phase 5: Memorize the Best Solution

Phase 6: Perform Scout Phase

Phase 7: Plot the result

Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \le x_1, x_2 \le 5$

MATLAB CODE: EXPLANATION

Phase I: Input parameter

```
%% Phase 1: Input parameters
FoodSource = 100;       % Food Sources Bees
D = 2;                  % Dimension of the problem
lb = [-5 -5];           % Lower bound of the variables
ub = [5 5];             % Upper bound of the variables
max_iter = 100;         % Maximum number of iteration
N = FoodSource./2;      % Population Size
limit = (N.*D);         % Used for Scoutt Phase
trial=zeros(N,1);       % Initialize to ZERO
```

Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \le x_1, x_2 \le 5$

MATLAB CODE: EXPLANATION

```
%% Phase 2: Defining the objective function
                & Fitness function
```

Defining Objective Function

```
function out = fns1(X)
    x1 = X(:,1);

    x2 = X(:,2);

    out = x1.^2-x1.*x2+x2.^2+2.*x1+4.*x2+3;

end
```

Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \le x_1, x_2 \le 5$

Maximize

$$\text{fit} = \begin{cases} \frac{1}{1+f} \; ; & f \ge 0 \\ 1 + |f| \; ; & f < 0 \end{cases}$$

MATLAB CODE: EXPLANATION

```
%%% Calculate fitness
function out = fns(X)
    x1 = X(:,1);
    x2 = X(:,2);
    fx = x1.^2-x1.*x2+x2.^2+2.*x1+4.*x2+3;
    for i=1:size(fx,1)
        if fx(i,:)>=0
            fit(i,:) = 1./(1+fx(i,:));
        elseif fx(i,:)<0
            fit(i,:) = 1+abs(fx(i,:));
        end
    end

    out = fit;
```

Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \leq x_1, x_2 \leq 5$

Use $x = L + rand.*(U - L)$

MATLAB CODE: EXPLANATION

Phase 3: Generate Initial Population

```
%% Phase 3: Generate Initial Population randomly

for i=1:N
    for j = 1:D
        pos(i,j) = lb(:,j) + rand.*(ub(:,j)-lb(:,j));
    end
end
```

Position

Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \leq x_1, x_2 \leq 5$

Use $x = L + rand.*(U - L)$

fns1

fns

MATLAB CODE: EXPLANATION

Phase 3: Generate Initial Population

```
%% Phase 3: Generate Initial Population randomly

for i=1:N
    for j = 1:D
        pos(i,j) = lb(:,j) + rand.*(ub(:,j)-lb(:,j));
    end
end
```


```
fx = fns1(pos);      %% Minimize Objective Function

fx1 = fns(pos);      %% Maximize Fitness Function
```


Illustrative Example

Minimize $f(X)$

$$= x_1^2 - x_1x_2 + x_2^2 + 2x_1 + 4x_2 + 3$$

where $-5 \leq x_1, x_2 \leq 5$

Use $x = L + rand.*(U - L)$

MATLAB CODE: EXPLANATION

Phase 3: Generate Initial Population

```
%% Phase 3: Generate Initial Population randomly

for i=1:N
    for j = 1:D
        pos(i,j) = lb(:,j) + rand.*(ub(:,j)-lb(:,j));
    end
end

fx = fns1(pos);      %% Minimize Objective Function

fx1 = fns(pos);      %% Maximize Fitness Function
```

Select update Variable & Partner

- Generate a new solution

$$X_{new} = X + \phi(X - X_p)$$

$\phi \in [-1,1]$;

- Calculate new fitness.

- Apply greedy selection

(accept new solution if it is better).

$f_{new} < f_{old}$;

Employed bee Phase:

Employed bee Phase:

Select the random variable to change:

Select the random partner:

Create a new food location:

Perform greedy selection:

Trial Value:

food Source

$$\begin{bmatrix} \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots \end{bmatrix}$$

Employed bee Phase:

1st employed bee: $[3.1472 \quad -4.0246]$

Select the random variable to change:

Let it be 1 : $[3.1472 \quad -4.0246]$

Select the random partner

Let partner be 4: $[4.1338 \quad 4.5751]$

food Source

$$\begin{bmatrix} 3.1472 & -4.0246 \\ 4.0579 & -2.2150 \\ -3.7301 & 0.4688 \\ 4.1338 & 4.5751 \\ 1.3236 & 4.6489 \end{bmatrix}$$

Maximize $f(x)$

$$\begin{bmatrix} 31.9645 \\ 32.6168 \\ 13.2971 \\ 48.6753 \\ 41.4537 \end{bmatrix}$$

Employed bee Phase:

Select the random variable to change:

Select the random partner:

Create a new food location:

Perform greedy selection:

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
for i=1:N
    p2c = ceil(rand*D);       % Choose the Variable to change
    partner = ceil(rand*N);   % Choose the Partner
```

Employed bee Phase:

Select the random variable to change:

Select the random partner:

$i = 1$

Create a new food location:

Perform greedy selection:

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
for i=1:N

    p2c = ceil(rand*D);       % Choose the Variable to change
    partner = ceil(rand*N);   % Choose the Partner

    while(partner==i)
        partner = ceil(rand*N);
    end
```

Employed bee Phase:

Select the random variable to change:

Select the random partner:

Create a new food location:

$$X_{new} = X + \phi(X - X_p)$$
$$\phi \in [-1,1];$$

Perform greedy selection:

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
for i=1:N
    Xnew = pos(i,:);          % For new solution "i" FoodSource
    p2c = ceil(rand*D);       % Choose the Variable to change
    partner = ceil(rand*N);   % Choose the Partner

    while(partner==i)
        partner = ceil(rand*N);
    end
```

Employed bee Phase:

1st employed bee: [3.1472 −4.0246]

Select the random variable to change:

Let it be 1 : [3.1472 −4.0246]

Select the random partner

Let partner be 4: [4.1338 4.5751]

Create a new food location:

Let $\phi = 0.71$

$X_{new} = X + \phi(X - X_p)$

$\phi \in [-1,1]$;

$X_{new} = 3.1472 + 0.71(3.1472 - 4.1338)$

$= 2.4467 \in (-5,5)$

Thus, $X_1 = [2.4467 \quad -4.0246]$

	food Source		Maximize $f(x)$	Minimize fit	trial
X →	3.1472	−4.0246	31.9645	0.0303	0
	4.0579	−2.1250	32.6168	0.0297	0
	−3.7301	0.4688	13.2971	0.0699	0
	4.1338	4.5751	48.6753	0.0201	0
	1.3236	4.6489	41.4537	0.0236	0

	food Source		Maximize $f(x)$	Minimize fit	trial
	.....	.....	.....	.....	...
	.....	.....	.....	.....	...
	.....	.....	.....	.....	...
	.....	.....	.....	.....	...
	.....	.....	.....	.....	...

Employed bee Phase:

Select the random variable to change:

Select the random partner :

Create a new food location:

$X_{new} = X + \phi(X - X_p)$

$\phi \in [-1,1]$;

Perform greedy selection:

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
for i=1:N
    Xnew = pos(i,:);          % For new solution "i" FoodSource
    p2c = ceil(rand*D);       % Choose the Variable to change
    partner = ceil(rand*N);   % Choose the Partner

    while(partner==i)
        partner = ceil(rand*N);
    end

    X = pos(i,p2c);
    Xp = pos(partner,p2c);
    Xnew(p2c) = X + (rand-0.5).*2.*(X-Xp);
```

Employed bee Phase:

Select the random variable to change:

Select the random partner :

Create a new food location:

$X_{new} = X + \phi(X - X_p)$

$\phi \in [-1,1]$;

Perform greedy selection:

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
Xnew(p2c) = X + (rand-0.5).*2.*(X-Xp);

    %%% Check the bounds
 for j = 1:D
    if Xnew(j)>ub(j)
        Xnew(j) = ub(j);
    elseif Xnew(j)<lb(j)
        Xnew(j) = lb(j);
    end
 end
```

Employed bee Phase:

Select the random variable to change:

Select the random partner :

Create a new food location:

$$X_{new} = X + \phi(X - X_p)$$
$$\phi \in [-1,1];$$

Perform greedy selection:

$f_{new} < f_{old}$; Minimization

Trial Value:

MATLAB CODE: EXPLANATION

Phase 4: Employed bee Phase:

```
Xnew(p2c) = X + (rand-0.5).*2.*(X-Xp);

%%% Perform Greedy Selection
 fnew = fns1(Xnew);
if fnew < fx(i,:)
    pos(i,:) = Xnew;
    fx(i,:)  = fnew;
    trial(i)=0;
else
    trial(i)=trial(i) + 1;
end
```

fns1 → Min

- Calculate probabilities

  Select the random number (r)

  If r < prob.

- New solution is generated with the help of partner solution (X_p) by

$$X_{new} = X + \phi(X - X_p)$$
$$\phi \in [-1,1];$$

- Perform greedy selection:

$$f_{new} < f_{old};$$

then Update the solution, otherwise NOT

Onlooker phase

MATLAB CODE: EXPLANATION

Calculate probabilities

Select the random number (r)

If r < prob.

```
%%%%% ONLOOKER BEE PHASE START %%%%%%%

prob = fx./sum(fx);

for i=1:N
    if(rand<prob(i))
```

MATLAB CODE: EXPLANATION

Calculate probabilities

Select the random number (r)

If r < prob.

- New solution is generated with the help of partner solution (X_p) by

$$X_{new} = X + \phi(X - X_p)$$

$\phi \in [-1,1]$;

```
%%%%% ONLOOKER BEE PHASE START %%%%%%%

prob = fx./sum(fx);

for i=1:N
    if(rand<prob(i))
        Xnew = pos(i,:);
        p2c = ceil(rand*D);
        partner = ceil(rand*N);
        while(partner==i)
            partner=ceil(rand*N);
        end
        X = pos(i,p2c);
        Xp = pos(partner,p2c);
        Xnew(p2c) = X + (rand-0.5).*2.*(X-Xp);
```


MATLAB CODE: EXPLANATION

Calculate probabilities

Select the random number (r)

If r < prob.

- New solution is generated with the help of partner solution (X_p) by

$$X_{new} = X + \phi(X - X_p)$$

$\phi \in [-1,1]$;

- Perform greedy selection:

$f_{new} < f_{old}$;

then Update the solution, otherwise NOT

```
    %%% Check the bounds
    for j = 1:D
        if Xnew(j)>ub(j)
            Xnew(j) = ub(j);
        elseif Xnew(j)<lb(j)
            Xnew(j) = lb(j);
        end
    end
%%% Perform Greedy Selection
fnew = fns1(Xnew);
if fnew < fx(i,:)
    pos(i,:) = Xnew;
    fx(i,:)  = fnew;
    trial(i)=0;
else
    trial(i)=trial(i)+1;
end
```

MATLAB CODE: EXPLANATION

Memorize the best answer

```
%% MEMORIZE THE BEST SOLUTION SO FAR

   [fxval, fxind] = min(fx);

   Gbest = pos(fxind,:);

   Xbest(iter,:) = Gbest;

   Fbest(iter) = fxval;

   Fbest1(iter) = fns1(Gbest);
```

Display Output

```
% Show Iteration Information

  disp(['Iteration ' num2str(iter) ':
        Best Cost = ' num2str(Fbest(iter))]);
```

MATLAB CODE: EXPLANATION

Scout Phase-

Find the abandoned solution (based on the value of *limit*)

- Generate a new solution randomly to replace them.

```
H = find(trial>limit);

if length(H)>0
%%%  Generate Population randomly for such "H"
for j = 1:D
    pos(H,j) = lb(:,j) + rand.*(ub(:,j)-lb(:,j));
end
fx(H,:)=fns1(pos(H,:));
end
```

MATLAB CODE: EXPLANATION

Phase 7: Plot the result

```
plot(Fbest1, 'r','LineWidth', 2);

xlabel('Iteration');

ylabel('Fitness Value');

grid on;
```