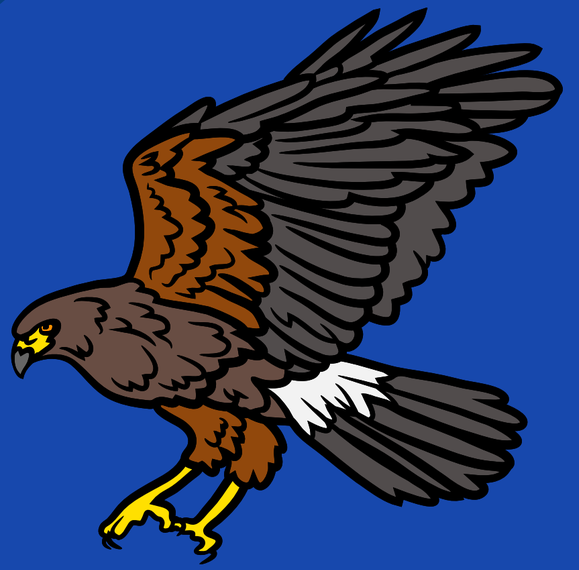
Harris Hawks Algoritm

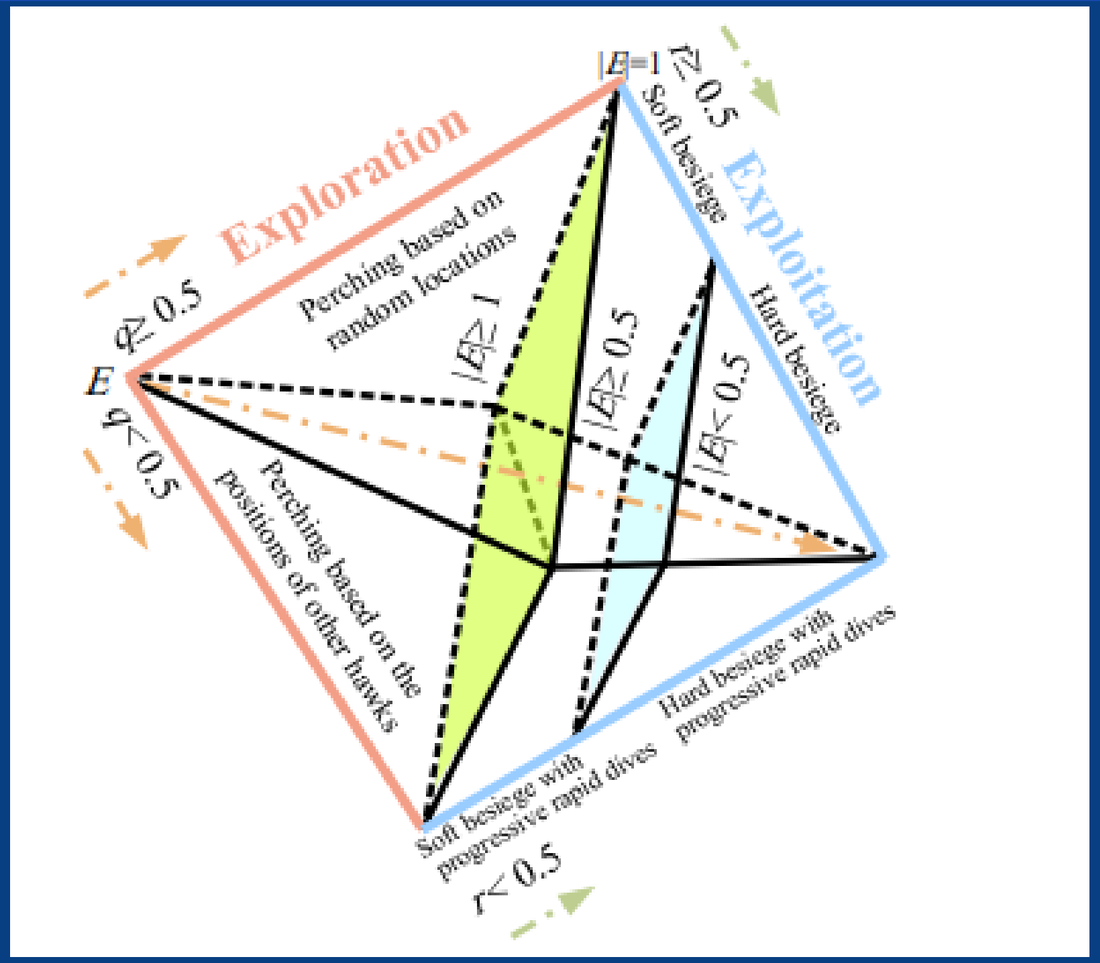
ALGORITMOS DE OPTIMIZACION

HALCON HARRIS HHO

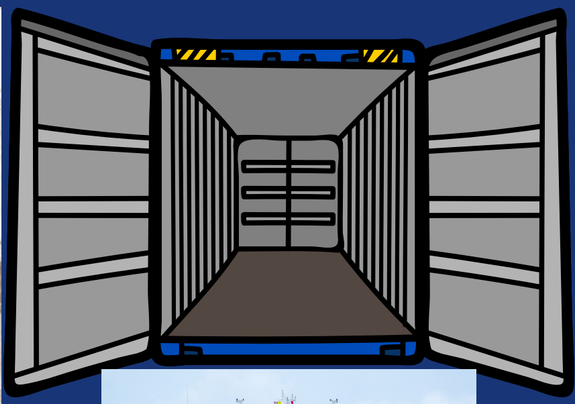


CONTENIDO

INTRODUCCIÓN



CONTEXTO



TECNICA HHO

Harris Hawks Algoritm

- Cooperación y búsqueda eficiente
- Adaptabilidad y diversificación
- Balance entre exploración y explotación
- Eficiencia computacional

COMO SE IMPLEMENTÓ

- Inicialización
- Evaluación de la aptitud
- Fases de exploración y explotación
- Actualización de la posición de los hawks
- Condiciones de parada

COMO SE IMPLEMENTÓ

Algorithm 1 Pseudo-code of HHO algorithm

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population $X_i (i = 1, 2, \ldots, N)$
while (stopping condition is not met) **do**
  Calculate the fitness values of hawks
  Set X_{rabbit} as the location of rabbit (best location)
  for (each hawk (X_i)) **do**
    Update the initial energy E_0 and jump strength J ▷ E_0=2rand()-1, J=2(1-rand())
    Update the E using Eq. (3)
    if $(|E| \geq 1)$ **then** ▷ Exploration phase
      Update the location vector using Eq. (1)
    if $(|E| < 1)$ **then** ▷ Exploitation phase
      if ($r \geq 0.5$ and $|E| \geq 0.5$) **then** ▷ Soft besiege
        Update the location vector using Eq. (4)
      else if ($r \geq 0.5$ and $|E| < 0.5$) **then** ▷ Hard besiege
        Update the location vector using Eq. (6)
      else if ($r < 0.5$ and $|E| \geq 0.5$) **then** ▷ Soft besiege with progressive rapid dives
        Update the location vector using Eq. (10)
      else if ($r < 0.5$ and $|E| < 0.5$) **then** ▷ Hard besiege with progressive rapid dives
        Update the location vector using Eq. (11)
Return X_{rabbit}

COMO SE IMPLEMENTÓ

```
clear all; close all; clc
load('Pesos_Productos.mat')

N = 50; % Número de agentes de búsqueda
T = 25; % Máximo número de iteraciones
lb = 1; % límite inferior para cada elemento de la solución
ub = 150; % límite superior para cada elemento de la solución
dim = numel(weight); % dimensión de la solución
iterations = 1:T; % Vector de números de iteración
bin_capacity = 150; % Define la capacidad de los bins
X = initialization(N, dim, ub, lb);
[Rabbit_Energy, Rabbit_Location, CNVG] = HHO(N, T, lb, ub, dim, @(sol) obj_func(sol, weight, bin_capacity));
plot(iterations, CNVG);
xlabel('Iteraciones');
ylabel('Cantidad de contenedores');
title('Número de iteraciones vs Cantidad de contenedores');

display(['The best location of HHO is: ', num2str(Rabbit_Location)]);
display(['The best fitness of HHO is: ', num2str(Rabbit_Energy)]);
```

COMO SE IMPLEMENTÓ

```
%%
function result = obj_func(sol, weight, bin_capacity) ...

%%

function [X]=initialization(N,dim,up,down) ...

%%
function [Rabbit_Energy,Rabbit_Location,CNVG]=HHO(N,T,lb,ub,dim,fobj) ...

%%
function o=Levy(d) ...
```

RESULTADOS

- Rabbit_Energy.
- Rabbit_Location.
- CNVG

COMO SE EXPLICAN LOS RESULTADOS

Soluciones Encontradas

Si las soluciones encontradas por HHO se acercan al óptimo global o superan otras soluciones conocidas, se considera que el algoritmo ha tenido un buen desempeño. Esto indica que HHO ha explorado y explotado eficientemente el espacio de búsqueda para encontrar soluciones de alta calidad

Analizar el Tiempo de Ejecución y la Eficiencia del Algoritmo

También se puede analizar el tiempo de ejecución y la eficiencia del algoritmo. Si HHO encuentra soluciones de alta calidad en un tiempo razonable, se considera que el algoritmo es eficiente y práctico para resolver el problema en cuestión

Borcelle School

FIN, MUCHAS GRACIAS

+56 9 59 633653

fernando.ramirezs@postgrado.uv.cl

www.linkedin.com