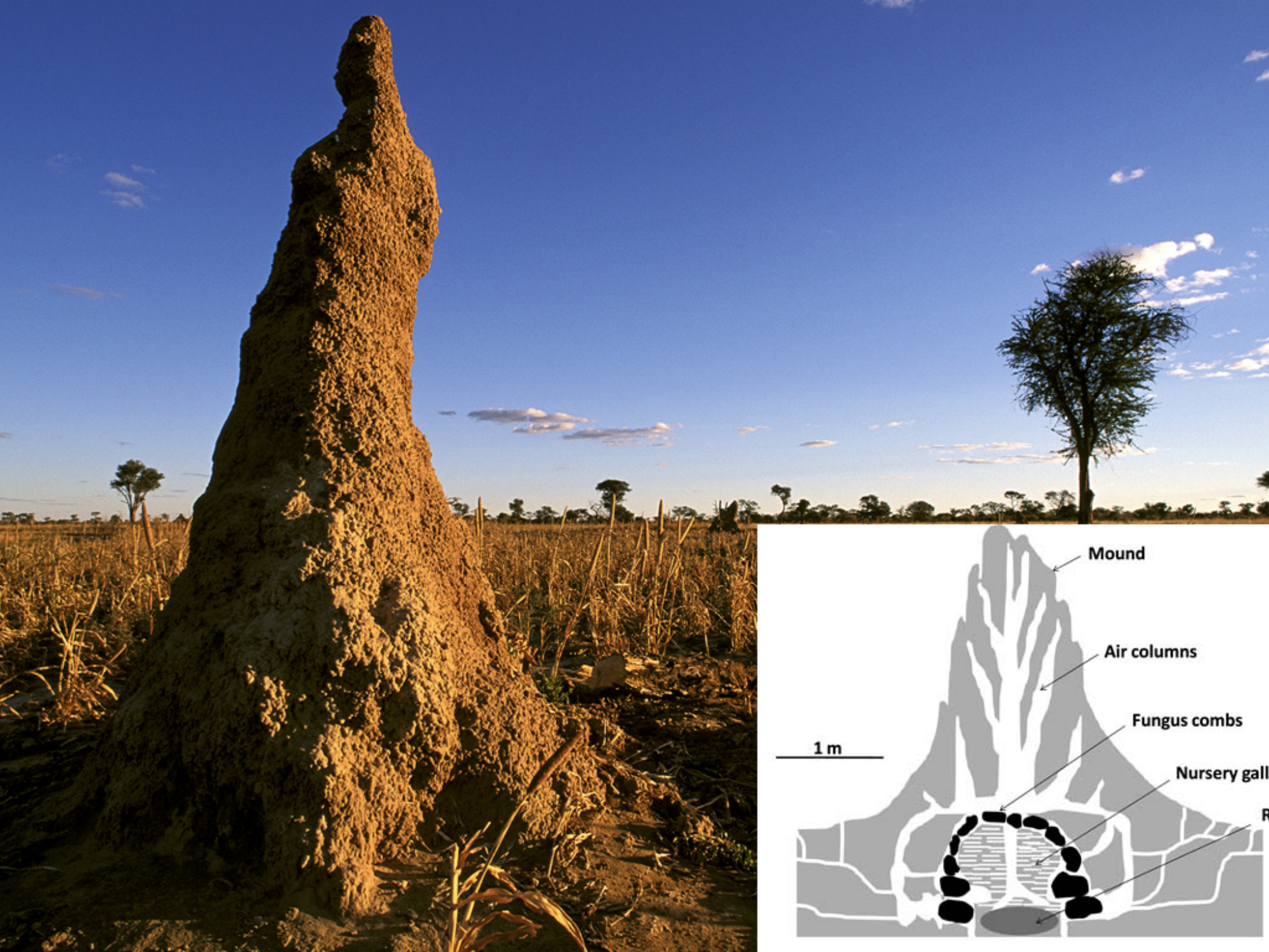
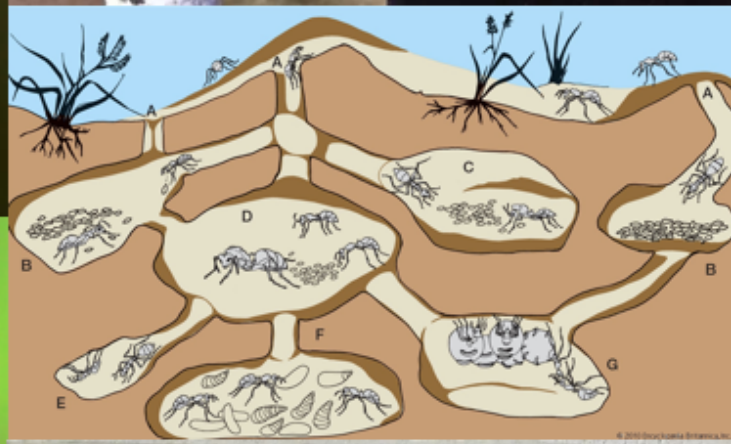
Inteligencia Computacional:

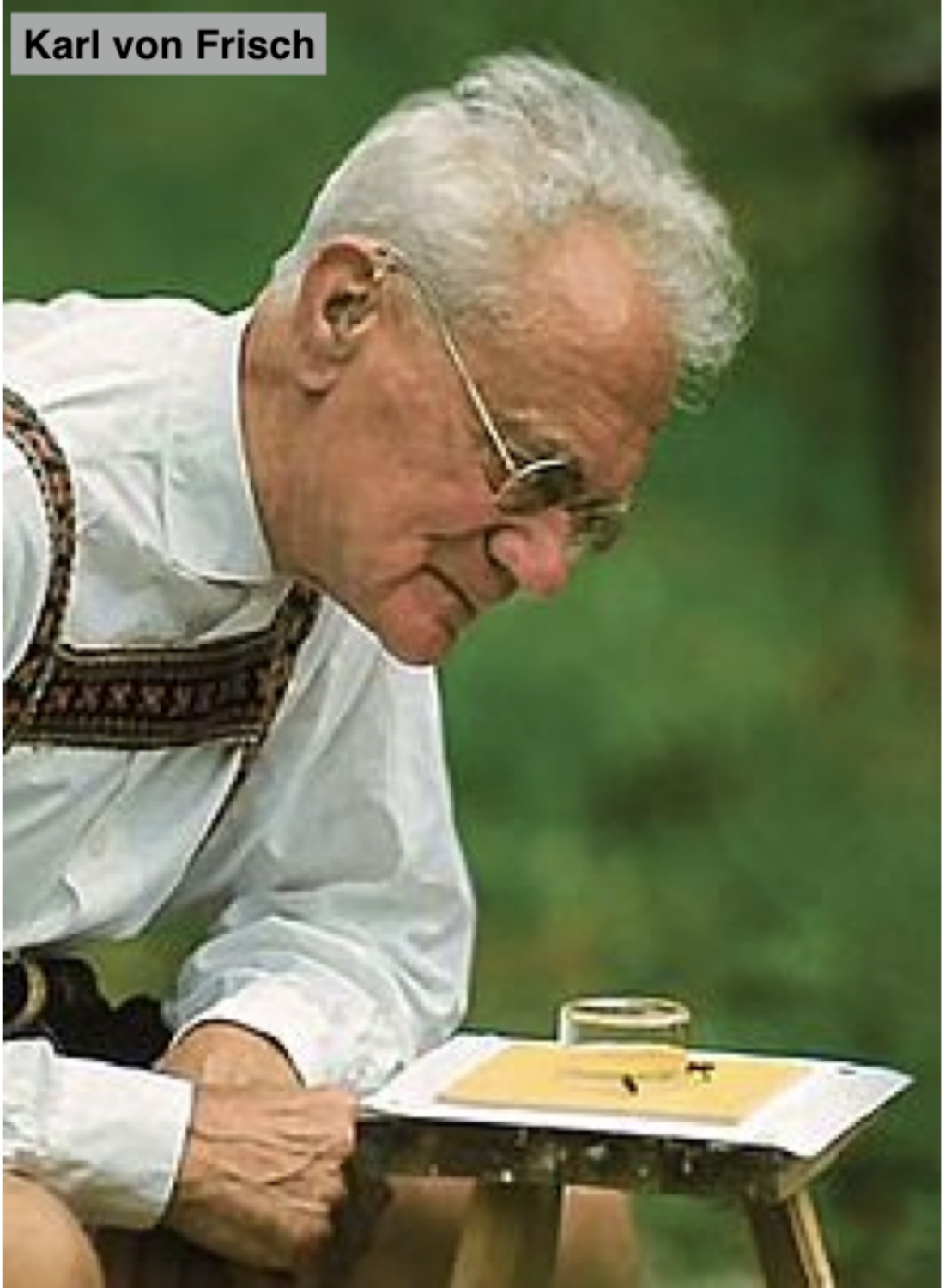
Inteligencia Colectiva

Dr. Gregorio Toscano
email: *gtoscano@cinvestav.com*

Introducción

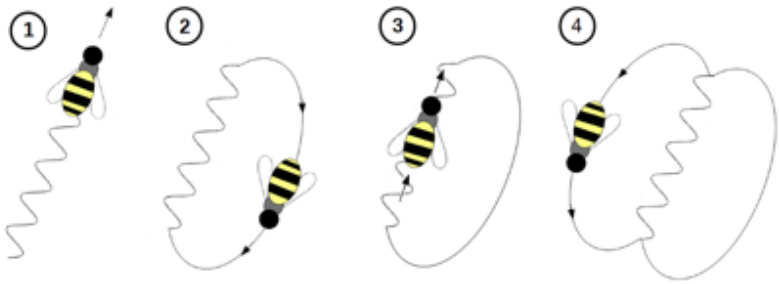

Mound
Air columns
Fungus combs
1 m
Nursery gall

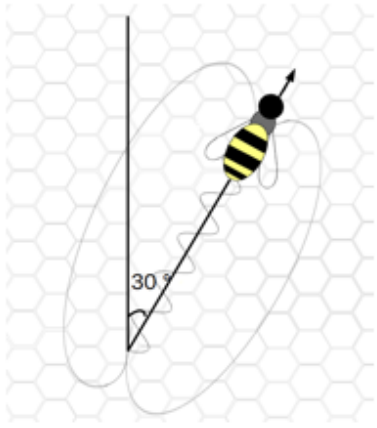

A
A
A
B
B
C
D
E
F
G

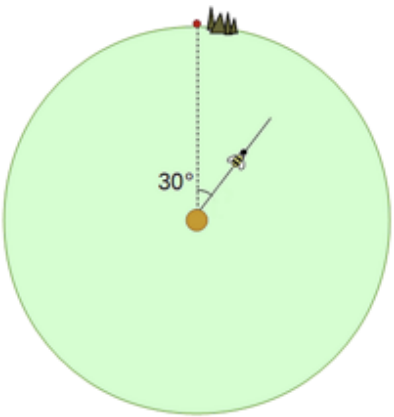


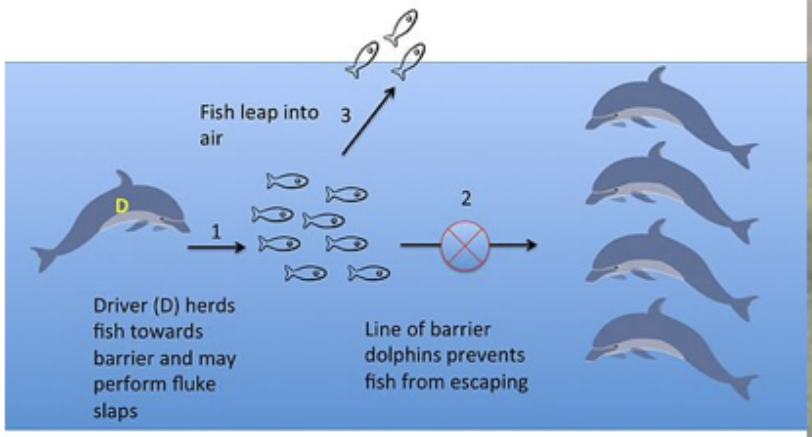
round dance

waggle dance








Fish leap into air
3
D
1
2
Driver (D) herds fish towards barrier and may perform fluke slaps
Line of barrier dolphins prevents fish from escaping

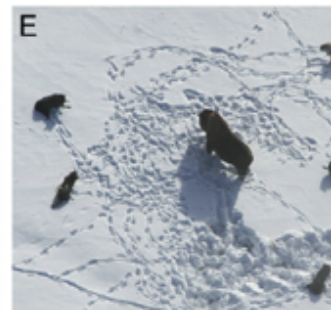

D


A


B


C


D

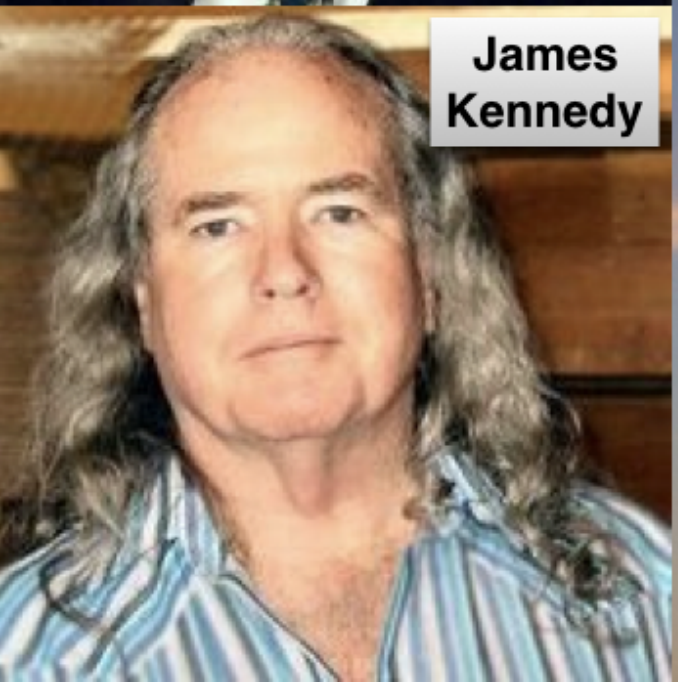

E

Objetivo de la inteligencia colectiva

Es modelar el comportamiento simple de individuos, y las interacciones locales con el ambiente y sus vecinos para obtener comportamientos más complejos que pueden ser usados para resolver problemas principalmente de optimización.

PSO


Russell
Eberhart


James
Kennedy

Particle Swarm Optimization

James Kennedy[1] and Russell Eberhart[2]

[1]Washington, DC 20212
kennedy_jim@bls.gov

[2]Purdue School of Engineering and Technology
Indianapolis, IN 46202-5160
eberhart@engr.iupui.edu

ABSTRACT

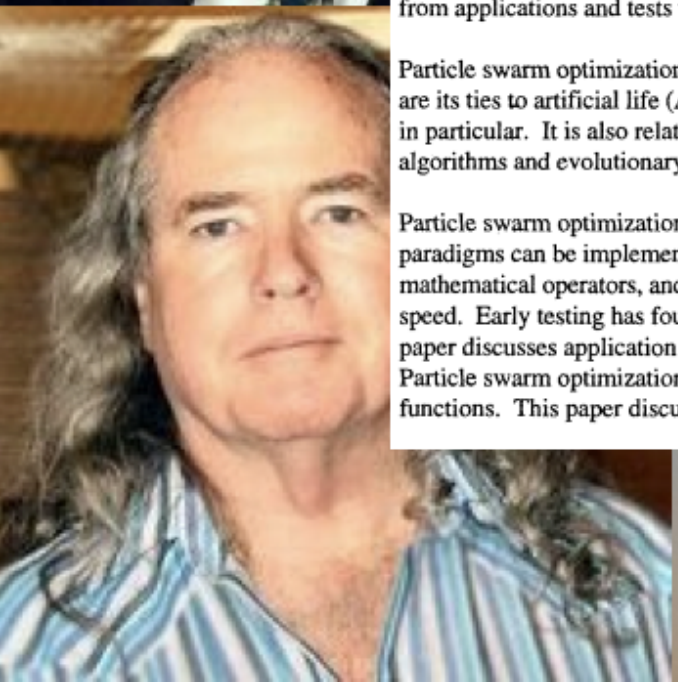
A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described.

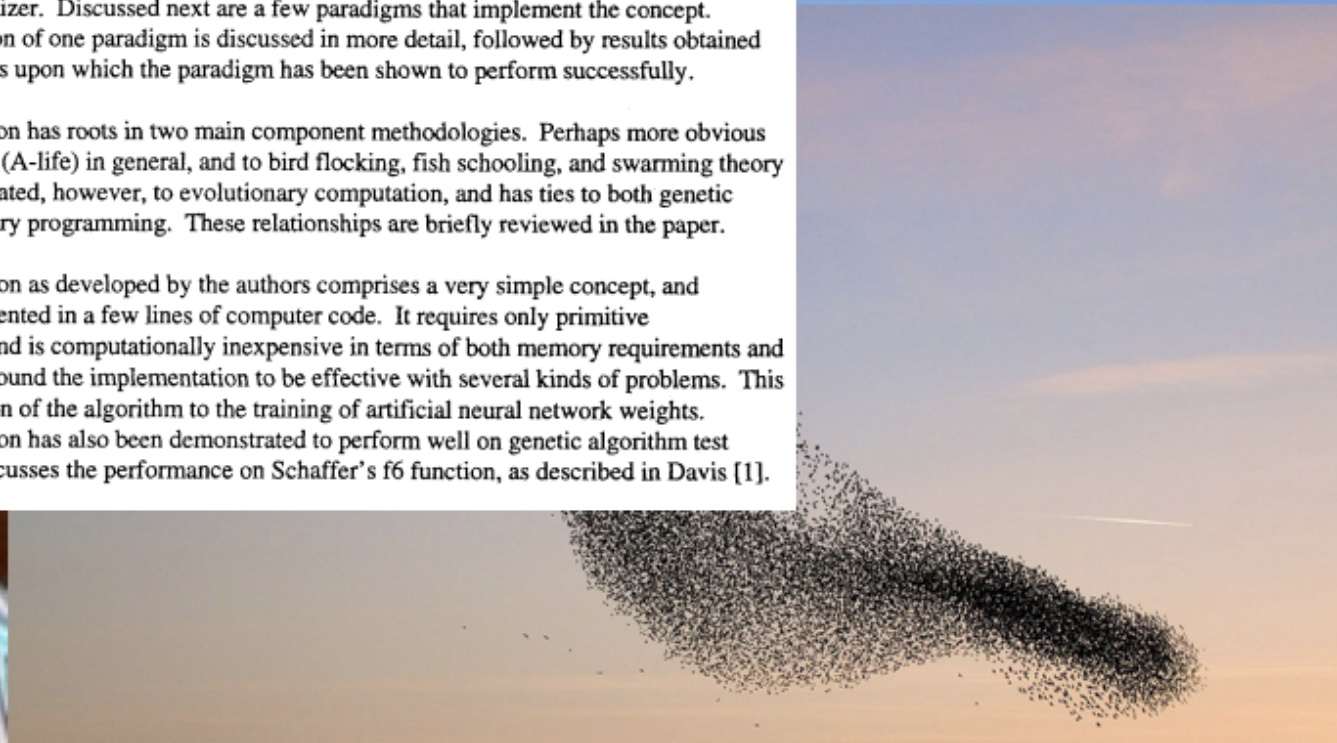
1 INTRODUCTION

This paper introduces a method for optimization of continuous nonlinear functions. The method was discovered through simulation of a simplified social model; thus the social metaphor is discussed, though the algorithm stands without metaphorical support. This paper describes the particle swarm optimization concept in terms of its precursors, briefly reviewing the stages of its development from social simulation to optimizer. Discussed next are a few paradigms that implement the concept. Finally, the implementation of one paradigm is discussed in more detail, followed by results obtained from applications and tests upon which the paradigm has been shown to perform successfully.

Particle swarm optimization has roots in two main component methodologies. Perhaps more obvious are its ties to artificial life (A-life) in general, and to bird flocking, fish schooling, and swarming theory in particular. It is also related, however, to evolutionary computation, and has ties to both genetic algorithms and evolutionary programming. These relationships are briefly reviewed in the paper.

Particle swarm optimization as developed by the authors comprises a very simple concept, and paradigms can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing has found the implementation to be effective with several kinds of problems. This paper discusses application of the algorithm to the training of artificial neural network weights. Particle swarm optimization has also been demonstrated to perform well on genetic algorithm test functions. This paper discusses the performance on Schaffer's f6 function, as described in Davis [1].





Particle Swarm Optimization

James Kennedy[1] and Russell Eberhart[2]

[1]Washington, DC 20212
kennedy_jim@bls.gov

Particle swarm optimization

View Document

12520 Paper Citations

23 Patent Citations

36017 Full Text Views

Related Articles

A new optimizer using particle swarm theory

A modified particle swarm optimizer

Genetic algorithms and simulated annealing: a marriage proposal

View All

2 Author(s)

J. Kennedy ; R. Eberhart

View All Authors

Abstract | Authors | Figures | References | Citations | Keywords | Metrics | Media

Abstract:
A concept for the optimization of nonlinear functions using particle swarm methodology is introduced. The evolution of several paradigms is outlined, and an implementation of one of the paradigms is discussed. Benchmark testing of the paradigm is described, and applications, including nonlinear function optimization and neural network training, are proposed. The relationships between particle swarm optimization and both artificial life and genetic algorithms are described.

Published in: Neural Networks, 1995. Proceedings., IEEE International Conference on

Date of Conference: 27 Nov.-1 Dec. 1995

INSPEC Accession Number: 5263228

Date Added to IEEE *Xplore*: 06 August 2002

DOI: 10.1109/ICNN.1995.488968

ISBN Information:

Publisher: IEEE

Equación de velocidad

$$v_i^j(t+1) = Wv_i^j(t) + C_1 r_1(t)(y_i^j(t) - x_i^j(t)) + C_2 r_2(t)(\hat{y}_i^j(t) - x_i^j(t))$$

$x_i^j(t)$ y $v_i^j(t)$ denotan respectivamente, la posición y velocidad de la partícula i, en su dimensión j, en el tiempo t. W es el peso de inercia, C_1 y C_2 son los coeficientes de los componentes cognitivo y social, respectivamente, y r_1, r_2 son números aleatorios en $[0, 1]$.

Además, y_i es la mejor posición encontrada por la partícula i y $\hat{y}$ es la mejor posición del vecindario de la partícula i (componente social).

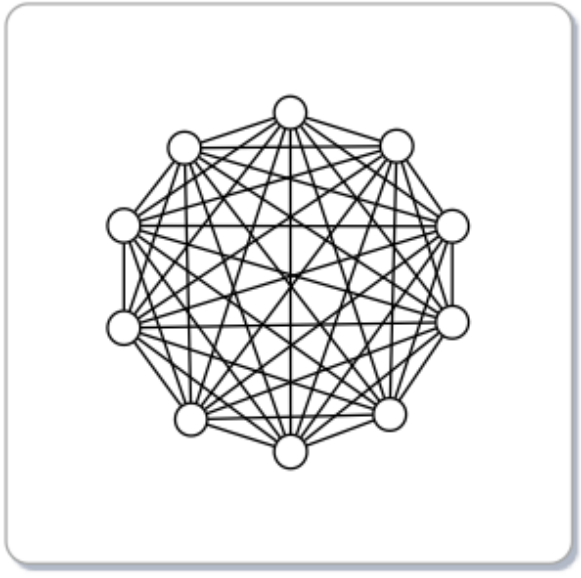
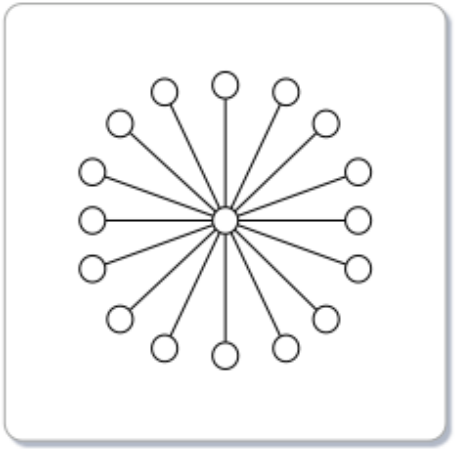
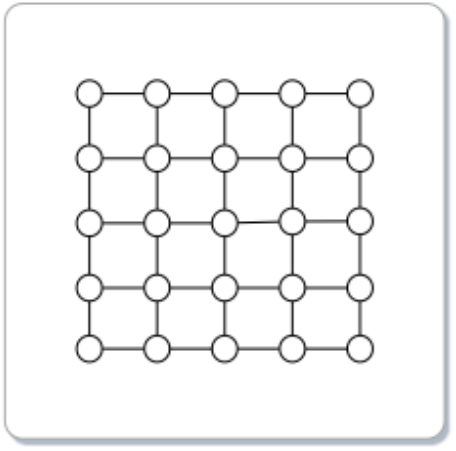


Figura 2.1: Topología completamente conectada.



(a) Estrella



(b) Malla

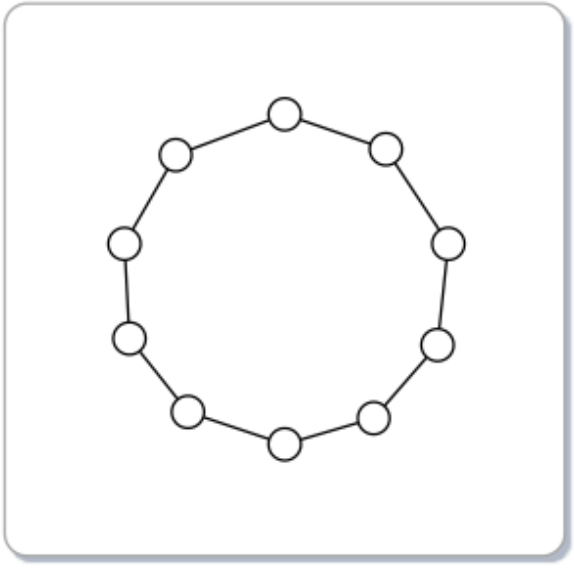
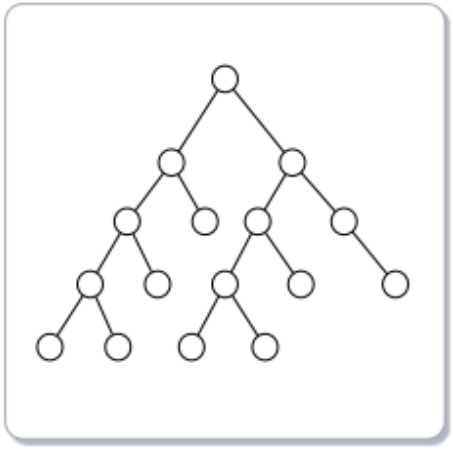
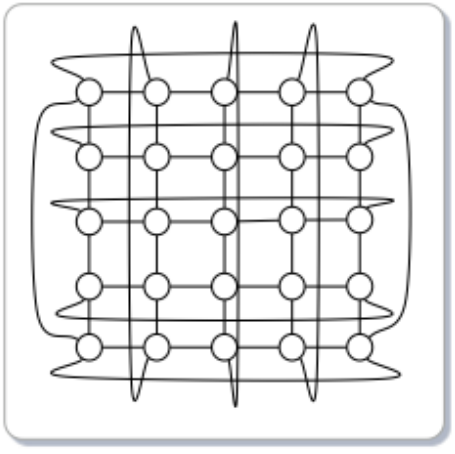


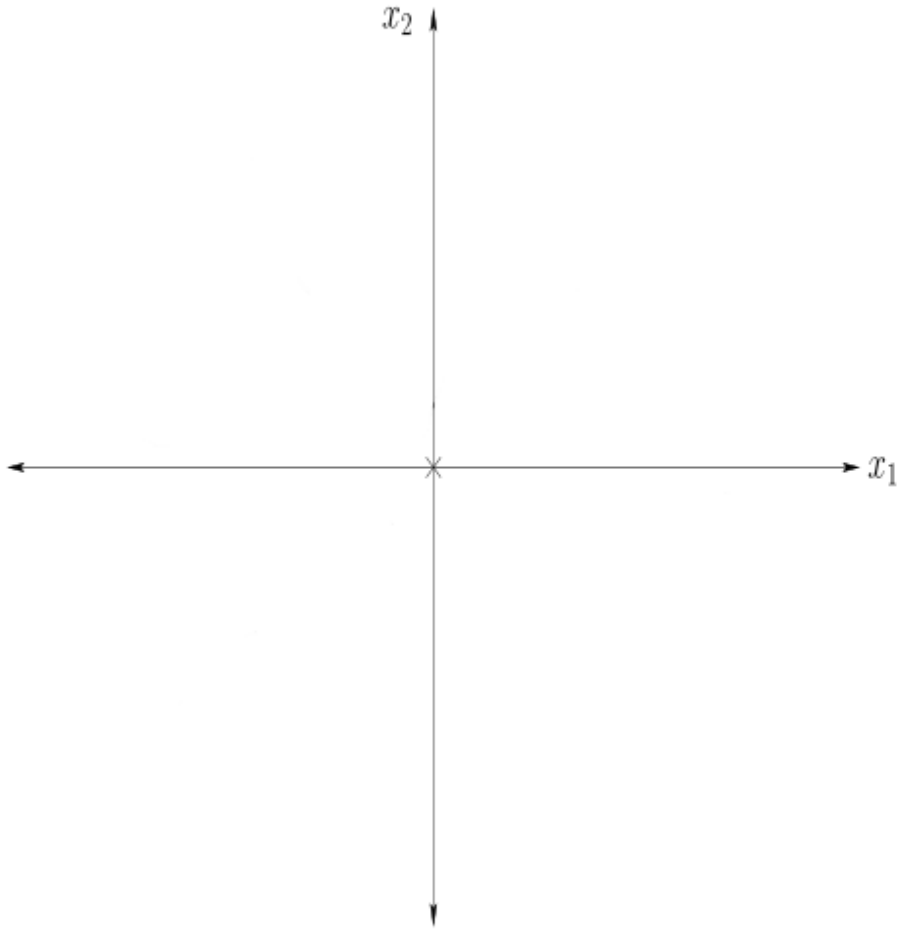
Figura 2.2: Topología en anillo.

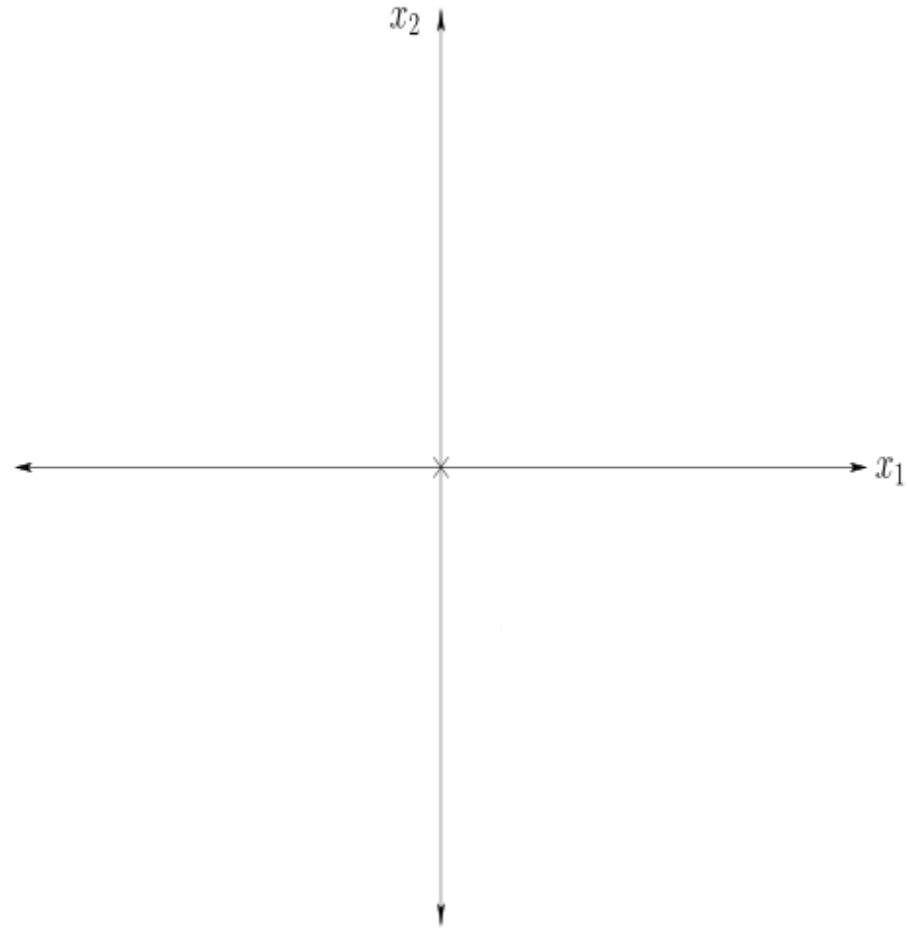


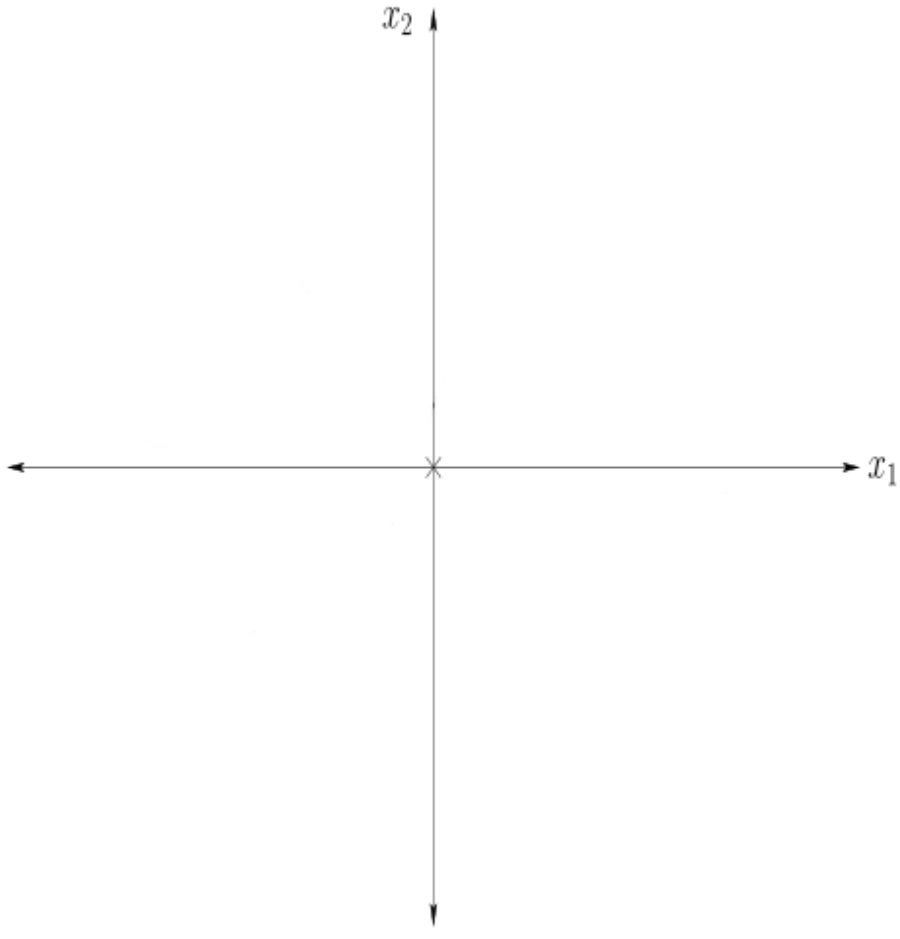
(c) Árbol

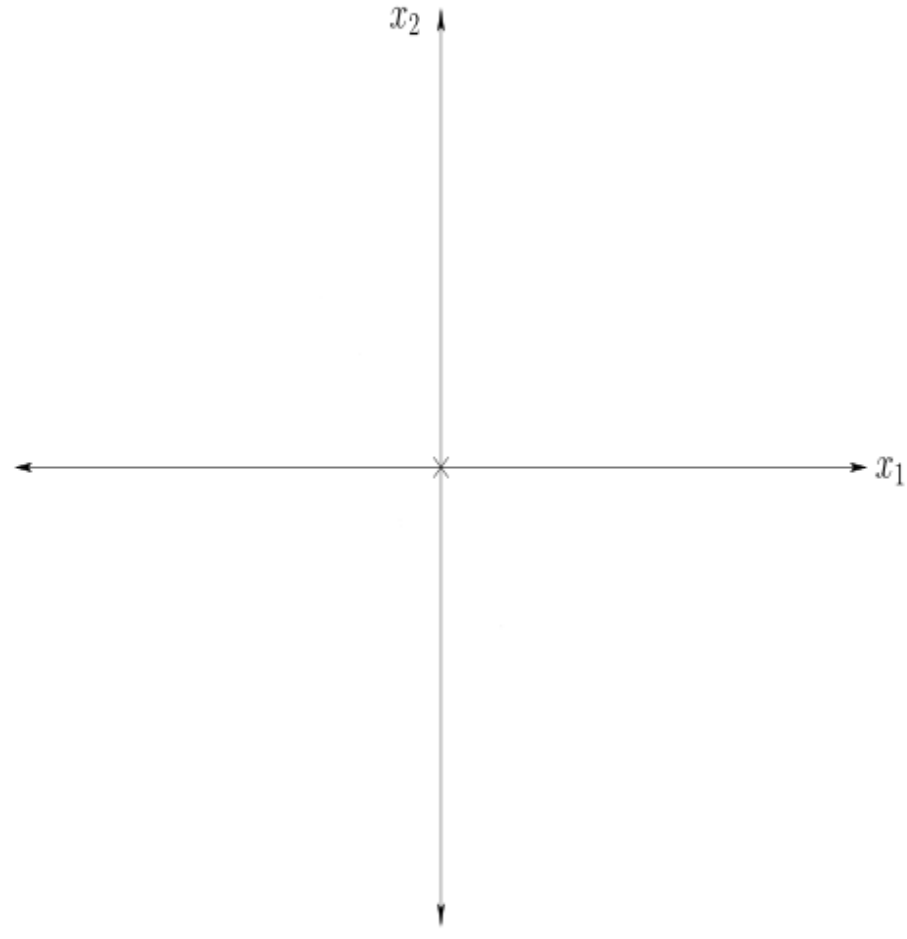


(d) Toroidal

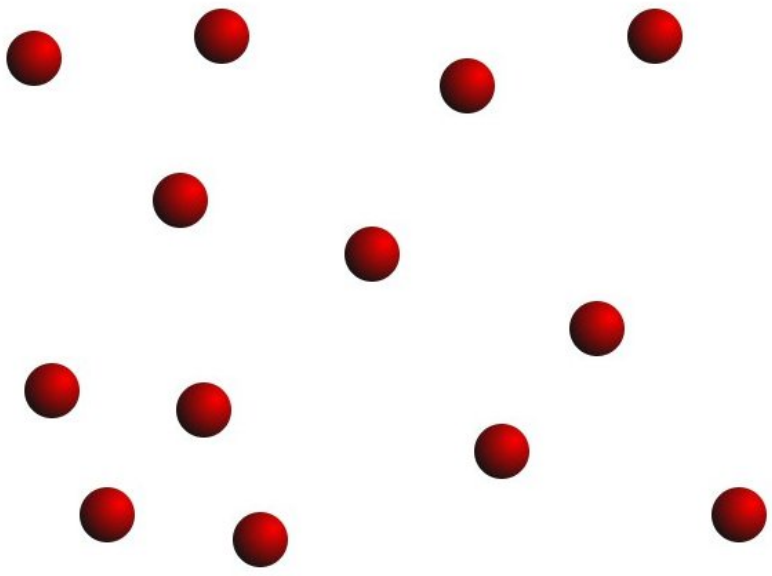

x_2
x_1


x_2
x_1


x_2
x_1


x_2
x_1

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

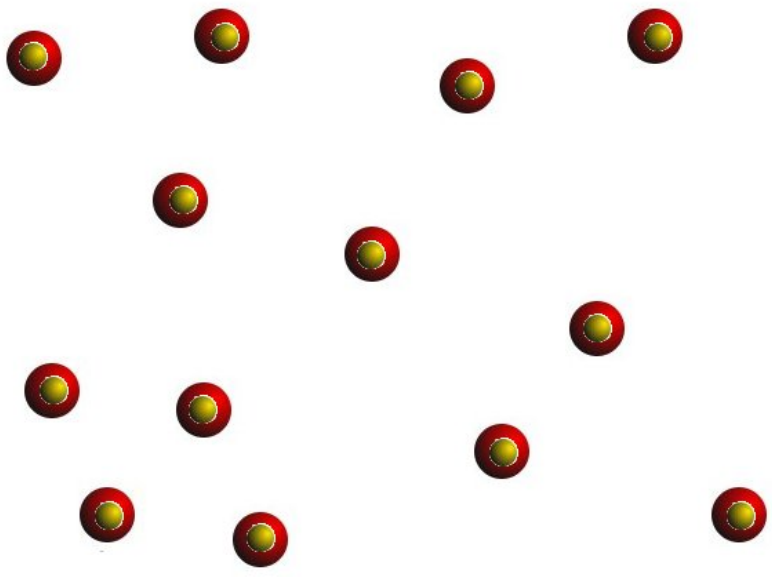
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

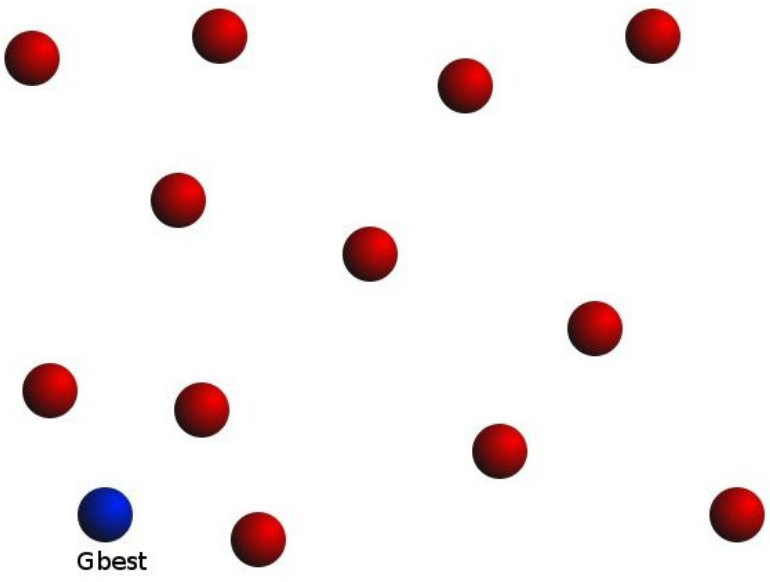
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

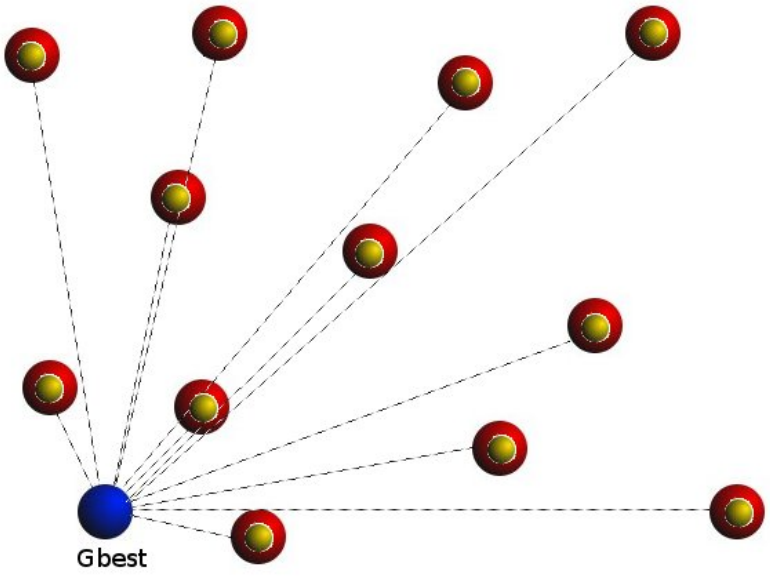
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

5. Compute its speed using:

$\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$

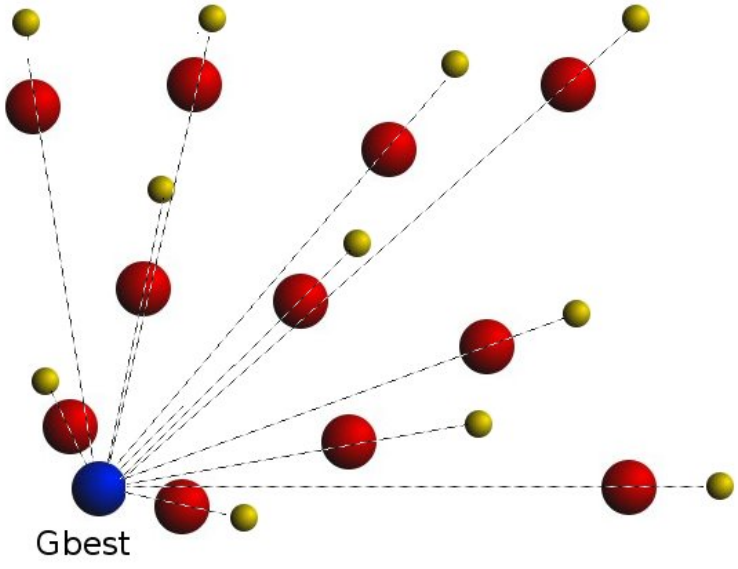
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

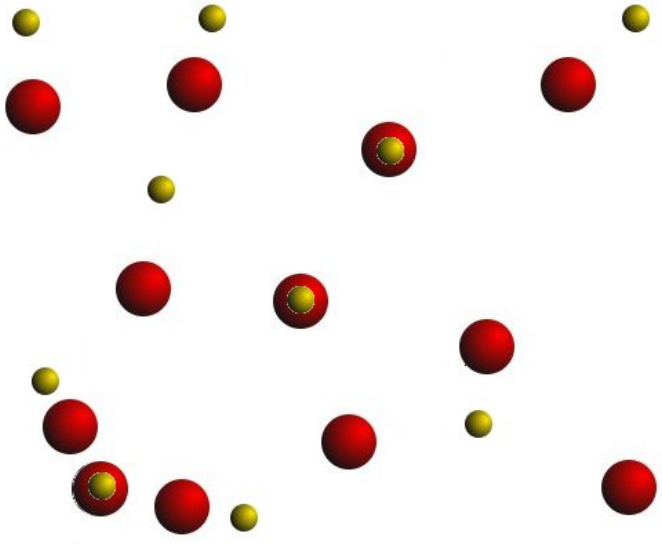
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

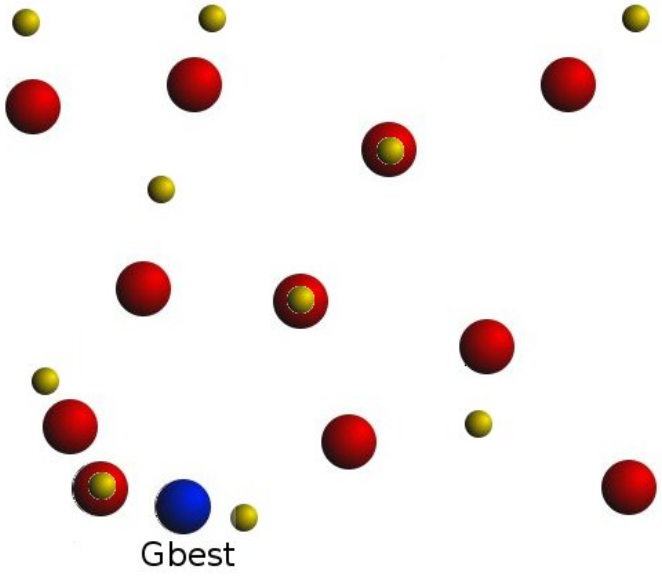
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

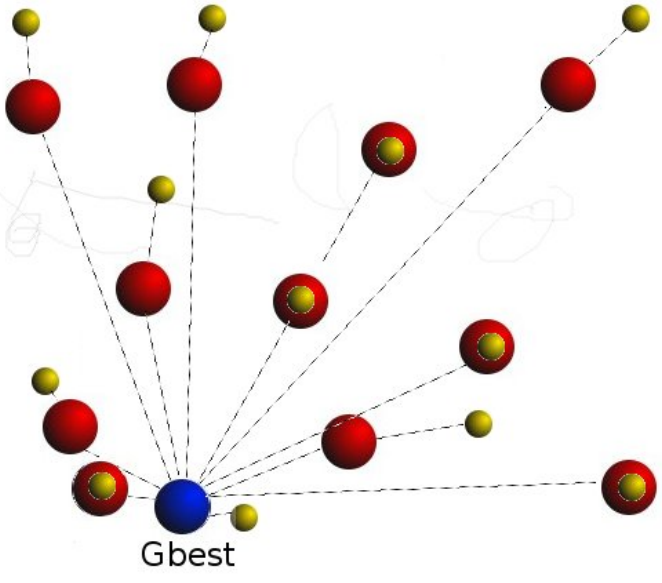
5. Compute its speed using:
   $\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Algoritmo del PSO



Begin

For each particle **do**

1. Initialize **x** and **v** randomly
2. Evaluate fitness function: $f(\mathbf{x})$
3. Initialize $\mathbf{y} \leftarrow \mathbf{x}$

EndFor

4. **Select** Gbest

Do

For each particle **x** **do**

5. Compute its speed using:
$\mathbf{v} \leftarrow W\mathbf{v} + C_1 r(0,1)(\mathbf{y} - \mathbf{x}) + C_2 r(0,1)(\hat{\mathbf{y}} - \mathbf{x})$
6. $\mathbf{x} \leftarrow \mathbf{x} + \mathbf{v}$ on each dimension
7. Evaluate the new position: $f(\mathbf{x})$
8. **If** $f(\mathbf{x}) \leq f(\mathbf{y})$ **then** $\mathbf{y} \leftarrow \mathbf{x}$
9. **If** $f(\mathbf{x}) \leq f(\hat{\mathbf{y}})$ **then** $\hat{\mathbf{y}} \leftarrow \mathbf{x}$

EndFor

While max. number of iterations is not reached

End.

Función esfera

$$f(x) = \sum_{i=1}^{n} x_i^2$$

Función de Ackley

$$f(x, y) = -20 \exp\left(-0.2\sqrt{0.5\left(x^2 + y^2\right)}\right)$$
$$- \exp\left(0.5\left(\cos\left(2\pi x\right) + \cos\left(2\pi y\right)\right)\right) + e + 20$$
$$x, y \in [-32.768, 32.768]$$

Otras funciones

https://en.wikipedia.org/wiki/Test_functions_for_optimization

Artículo original

http://doi.org/10.1109/ICNN.1995.488968

¿Preguntas?, ¿comentarios?:

gtoscano@cinvestav.mx