Particle Swarm Optimization

Prakash Kotecha
Debasis Maharana & Remya Kommadath
Department of Chemical Engineering
Indian Institute of Technology Guwahati

Swarm Intelligence

- “Any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies”*

- Examples of Swarms
  - bees swarming around their hive
  - ant colony with individual agents as ants
  - flock of birds is a swarm of birds
  - immune system is a swarm of cells
  - crowd is a swarm of people

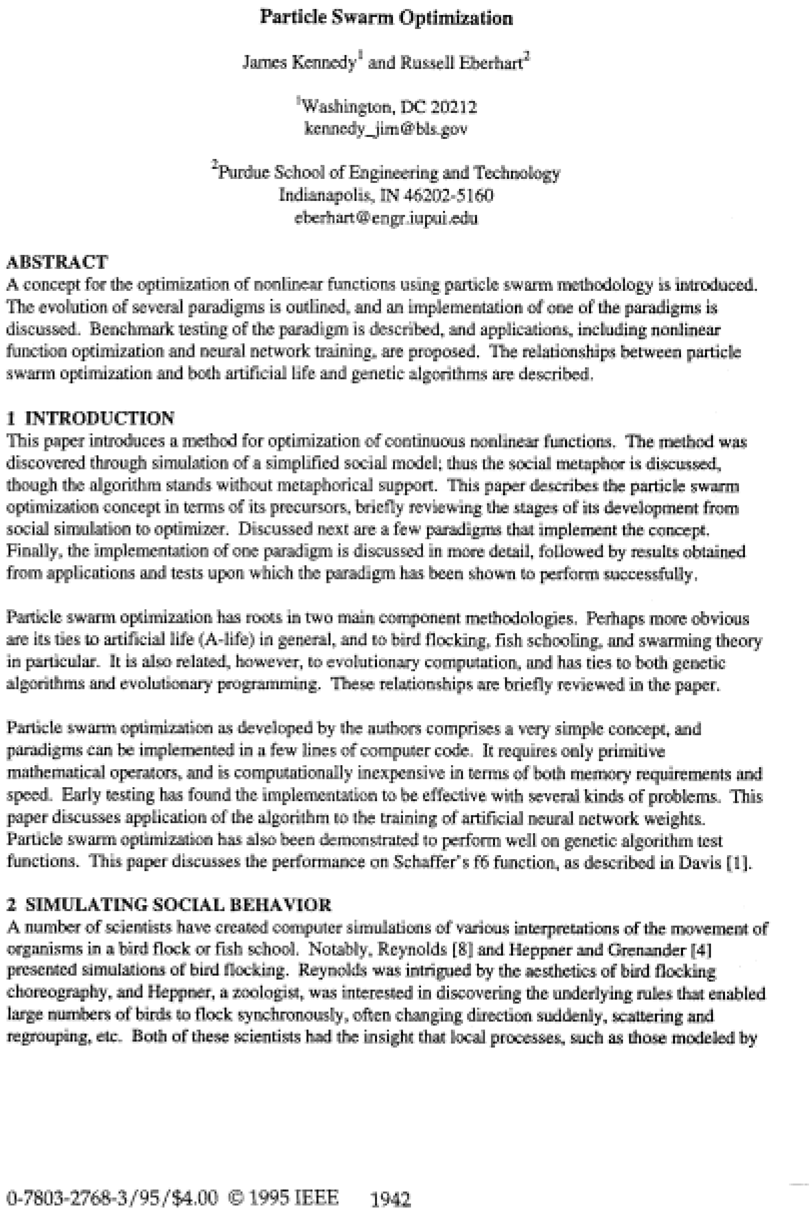
- Properties of swarm intelligent behaviour
  - self-organization
    - interactions are executed on the basis of purely local information without any relation to the global pattern
    - positive feedback, negative feedback, fluctuations and multiple interactions
  - division of labour
    - tasks performed simultaneously by specialized individuals

Swarm Intelligence: From Natural to Artificial Systems, New York, NY: Oxford University Press, 1999

An idea based on honey bee swarm for numerical optimization, Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

Particle Swarm Optimization

Proposed by J. Kennedy and R. Eberhart in 1995



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.

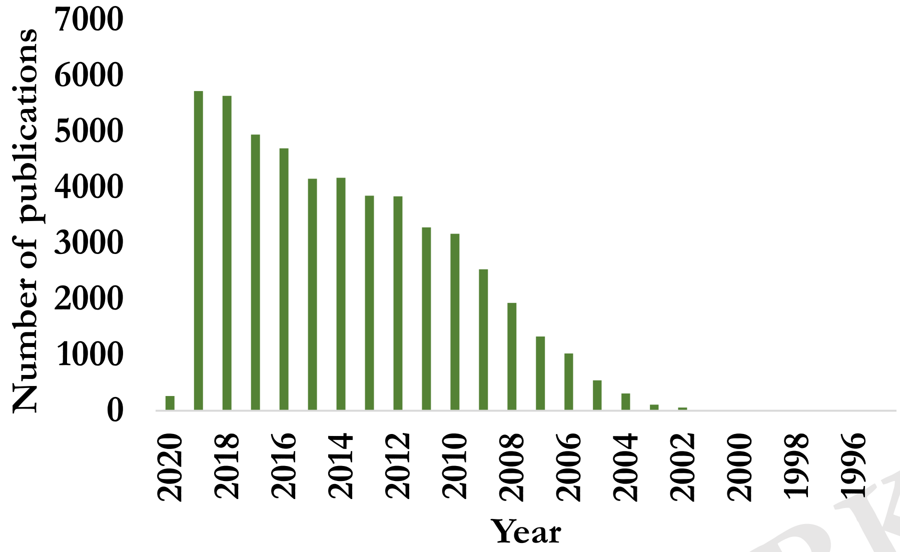
Particle swarm optimization as developed by the authors comprises a very simple concept, and paradigms can be implemented in a few lines of computer code. It requires only primitive mathematical operators, and is computationally inexpensive in terms of both memory requirements and speed. Early testing 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].

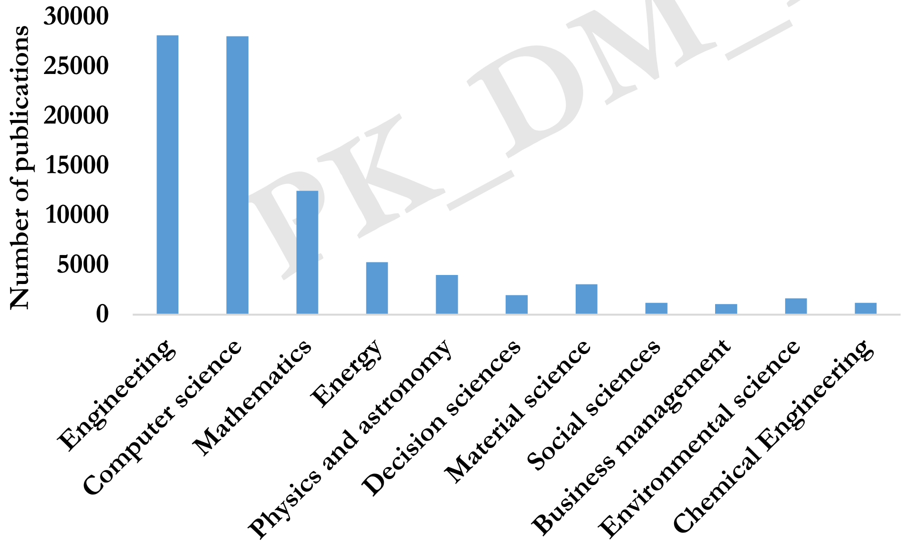
2 SIMULATING SOCIAL BEHAVIOR
A number of scientists have created computer simulations of various interpretations of the movement of organisms in a bird flock or fish school. Notably, Reynolds [8] and Heppner and Grenander [4] presented simulations of bird flocking. Reynolds was intrigued by the aesthetics of bird flocking choreography, and Heppner, a zoologist, was interested in discovering the underlying rules that enabled large numbers of birds to flock synchronously, often changing direction suddenly, scattering and regrouping, etc. Both of these scientists had the insight that local processes, such as those modeled by

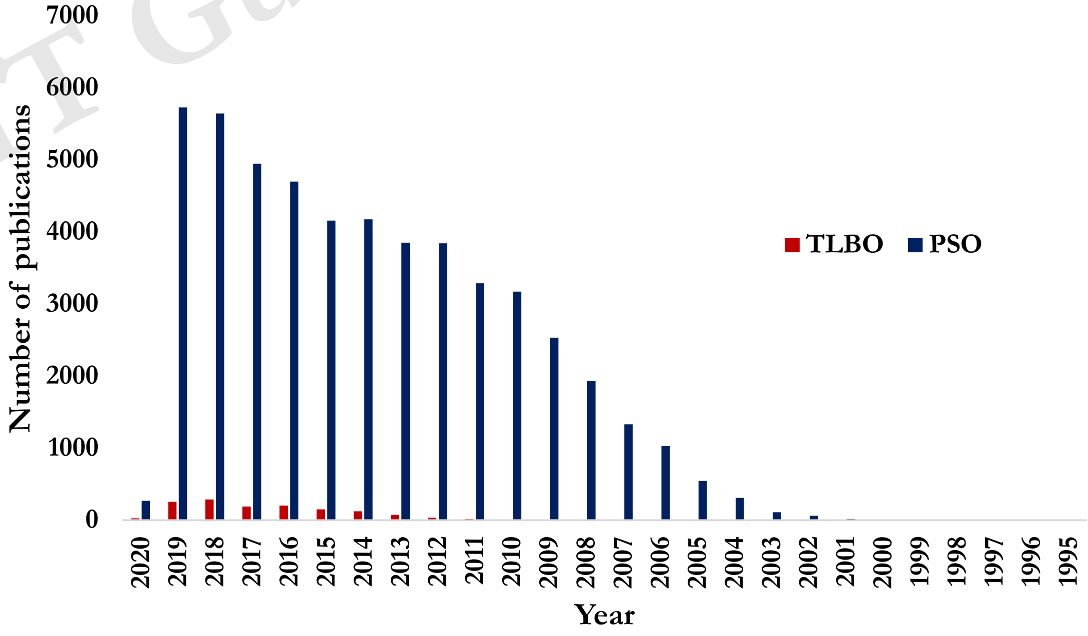
0-7803-2768-3/95/$4.00 © 1995 IEEE 1942

Particle Swarm Optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Australia, 1942-1948 vol.4, 1995

Particle Swarm Optimization







Particle Swarm Optimization (PSO)

- Models the social behaviour of bird flocking or fish schooling.
- Each particle/bird has a position and velocity associated with it.
- Particles change the position by adjusting their velocity to
  - seek food
  - avoid predators
  - identify optimized environmental parameters
- Each particle memorizes the best location identified by it.
- Particles communicate the information regarding the best location explored by them.
- Velocity of the particles are modified by using
  - flying experience of the particle
  - flying experience of the group

Particle Swarm Optimization (PSO)

- Initial position and velocity of particles are generated randomly within the search space.

- Particle velocity (v) is determined as

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$

- Position of a particle is modified as

$$X_i = X_i + v_i$$

v_i	Velocity of the i^{th} particle
w	Inertia of the particles
c_1 and c_2	Acceleration coefficients
r_1 and r_2	Random numbers $\in [0,1]$ of size ($1 \mathrm{x} D$)
$p_{best,i}$	Personal best of i^{th} particle
g_{best}	Global best
X_i	Position of i^{th} particle

- Evaluate the objective function f_i and update the population, irrespective of the fitness

- Update p_{best} and g_{best}

$$\left.\begin{array}{l} p_{best,i} = X_i \\ f_{p_{best,i}} = f_i \end{array}\right\} if \;\; f_i < f_{p_{best,i}}$$

$$\left.\begin{array}{l} g_{best} = p_{best,i} \\ f_{g_{best}} = f_{p_{best,i}} \end{array}\right\} if \;\; f_{p_{best,i}} < f_{g_{best}}$$

Velocity of a particle

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$

wv_i

Momentum part

- Serves as the memory of previous flight
- Prevent the particle from drastically changing the direction
- Biased towards the previous direction

$c_1 r_1 \left(p_{best,i} - X_i \right)$

Cognitive part

- Quantifies the performance of i^{th} particle relative to its past performance
- Particles are drawn back to their own best position
- Nostalgia of the particle

$c_2 r_2 \left(g_{best} - X_i \right)$

Social part

- Quantifies the performance of i^{th} particle relative to neighbors
- Particles are drawn towards the best position determined by the group
- Resembles the group norm that each particle seek to attain

A. P. Engelbrecht, Particle Swarm Optimization, *Computational Intelligence: An Introduction, Second Edition, chapter 16*, 2007

Possible cases

Cases	Better than its Personal best	Better than Global best	Remarks
Case 1	✖	✖	Do not update p_{best} and g_{best}
Case 2	✓	✖	Update p_{best} Do not update g_{best}
Case 3	✓	✓	Update p_{best} and g_{best}
Case 4 (Cannot happen)	✖	✓	Do not update p_{best} or g_{best}

$$\min \quad f(x) = \sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 10, \quad i = 1,2,3,4$$

Case 1:

Let X = [5 6], f = 61

p_{best} = [4 5], f_{pbest} = 41

g_{best} = [2 3], f_{gbest} = 13

$f > f_{pbest}$

$f > f_{gbest}$

Case 2:

Let X = [4 3], f = 25

p_{best} = [4 5], f_{pbest} = 41

g_{best} = [2 3], f_{gbest} = 13

$f < f_{pbest}$

f_{pbest} = 25 and p_{best} = [4 3]

$f > f_{gbest}$

Case 3:

Let X = [1 3], f = 10

p_{best} = [4 5], f_{pbest} = 41

g_{best} = [2 3], f_{gbest} = 13

$f < f_{pbest}$

f_{pbest} = 10 and p_{best} = [1 3]

$f < f_{gbest}$

f_{gbest} = 10 and g_{best} = [1 3]

Working of PSO: Sphere Function

Consider $\min \quad f(x) = \sum_{i=1}^{4} x_i^2; \quad 0 \le x_i \le 10, \quad i = 1, 2, 3, 4$

$f(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2$

Decision variables: x_1, x_2, x_3 and x_4

- Step 1: Fix population size, inertia weight, acceleration coefficient, maximum iterations

$$N_P = 5,\ w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$$

- Step 2: Generate random positions within the domain. Evaluate the fitness.

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

Determine personal and global best solutions

- Step 3: Generate random velocities within the domain of the variables.

$$P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \quad f = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix} \qquad v = \begin{bmatrix} 9 & 6 & 1 & 8 \\ 5 & 1 & 3 & 0 \\ 7 & 4 & 1 & 4 \\ 3 & 0 & 2 & 1 \\ 1 & 6 & 8 & 7 \end{bmatrix}$$

- Step 4: Determine the personal and global best of all the solutions.

$$P_{best} = P = \begin{bmatrix} 4 & 0 & 0 & 8 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f_{pbest} = \begin{bmatrix} 80 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix} \qquad g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{gbest} = 35$$

First Solution: Generation

$w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$

- Step 5: Generate two sets of random vectors

$r_1 = [0.4\ \ 0.3\ \ 0.9\ \ 0.5\ ]$ $\qquad r_2 = [0.8\ \ 0.2\ \ 0.7\ \ 0.4]$

$$v_1 = \begin{bmatrix} 9 & 6 & 1 & 8 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f = 80$$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \qquad f_{pbest_1} = 80$$

$$gbest = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \qquad f_{gbest} = 35$$

- Step 6: Determine velocity

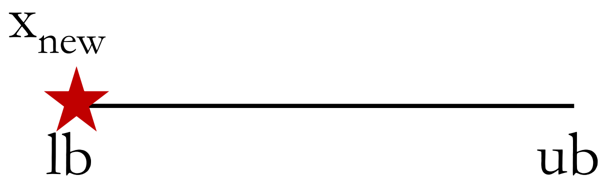
$$v_1 = 0.7 \text{ x } [9\ \ 6\ \ 1\ \ 8] +$$
$$1.5 \text{ x } [0.4\ \ 0.3\ \ 0.9\ \ 0.5\ ] \text{ x } ([4\ 0\ 0\ 8] - [4\ 0\ 0\ 8]) +$$
$$1.5 \text{ x } [0.8\ \ 0.2\ \ 0.7\ \ 0.4] \text{ x } ([0\ 3\ 1\ 5] - [4\ 0\ 0\ 8])$$
$$v_1 = [1.5\ \ 5.1\ \ 1.75\ \ 3.8]$$

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$
$$X_i = X_i + v_i$$

- Step 7: Determine position

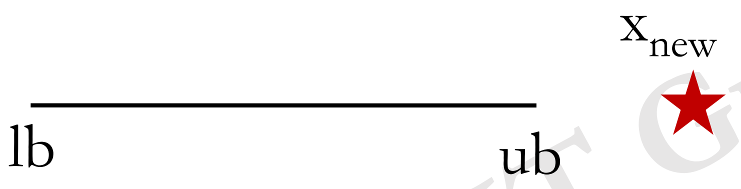
$$X_1 = [4\ 0\ 0\ 8] + [1.5\ \ 5.1\ \ 1.75\ \ 3.8]$$
$$= [5.5\ \ 5.1\ \ 1.75\ \ 11.8]$$

Bounding of solution

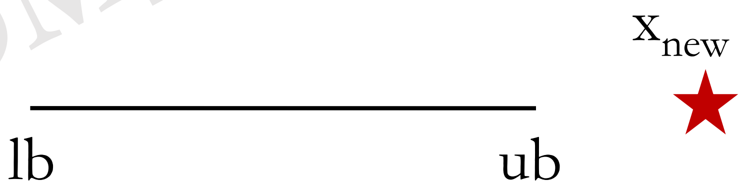


x_{new} is within bounds

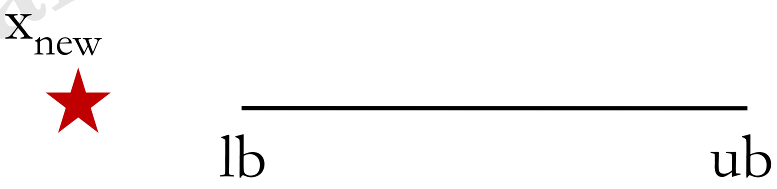
No bounding required



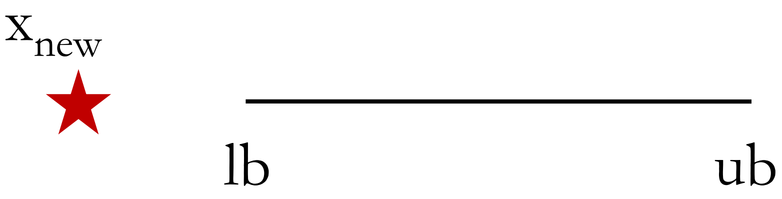
x_{new} violates the upper bound



Shift x_{new} to upper bound



x_{new} violates the lower bound



Shift x_{new} to lower bound

First Solution: Updating

- Step 8: Check bounds, bound for violation $\boxed{0 \le x_i \le 10}$

$$v_1 = \begin{bmatrix} 1.5 & 5.1 & 1.75 & 3.8 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_1 = 80$$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_{pbest_1} = 80$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \quad f_{gbest} = 35$$

$$X_1 = [5.5 \;\; 5.1 \;\; 1.75 \;\; 11.8] \qquad X_1 = [5.5 \;\; 5.1 \;\; 1.75 \;\; 10]$$

- Step 9: Evaluate fitness $f_1 = 5.5^2 + 5.1^2 + 1.75^2 + 10^2 = 159.32$

- Step 10: Update population

$$Pop = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3 & 1 & 9 & 7 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 140 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

- Step 11: No update of $p_{best,1}$ as new solution is not better. $p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_{pbest_1} = 80$

- Step 12: No update in g_{best} as new solution is not better. $g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \quad f_{gbest} = 35$

Second Solution: Generation

$w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$

- Step 1: Generate two sets of random vectors

$r_1 = [0.1 \;\; 0.4 \;\; 0.6 \;\; 0.3] \qquad r_2 = [0.7 \;\; 0.5 \;\; 0.8 \;\; 0.2]$

$$v_2 = \begin{bmatrix} 5 & 1 & 3 & 0 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} 3 & 1 & 9 & 7 \end{bmatrix} \qquad f = 140$$

$$p_{best,2} = \begin{bmatrix} 3 & 1 & 9 & 7 \end{bmatrix} \quad f_{pbest_2} = 140$$

$$g_{best} = \begin{bmatrix} 0 & 3 & 1 & 5 \end{bmatrix} \quad f_{gbest} = 35$$

- Step 2: Determine velocity

$$v_2 = 0.7 \text{ x } [5\ 1\ 3\ 0] + 1.5 \text{ x } [0.1 \;\; 0.4 \;\; 0.6 \;\; 0.3] \text{ x } ([3\ 1\ 9\ 7] - [3\ 1\ 9\ 7]) + 1.5 \text{ x } [0.7 \;\; 0.5 \;\; 0.8 \;\; 0.2] \text{ x } ([0\ 3\ 1\ 5] - [3\ 1\ 9\ 7])$$

$$v_2 = [0.35 \;\; 2.2 \;\; -7.5 \;\; -0.6]$$

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$

$$X_i = X_i + v_i$$

- Step 3: Determine position

$$X_2 = [3\ 1\ 9\ 7] + [0.35 \;\; 2.2 \;\; -7.5 \;\; -0.6] = [3.35 \;\; 3.2 \;\; 1.5 \;\; 6.4]$$

Second Solution: Updating

- Step 4: Check bounds, bound if violation $\boxed{0 \le x_i \le 10}$

$$v_2 = [0.35 \quad 2.2 \quad -7.5 \quad -0.6]$$

$$X_2 = [3 \quad 1 \quad 9 \quad 7] \qquad f_2 = 140$$

$$p_{best,2} = [3 \quad 1 \quad 9 \quad 7] \qquad f_{pbest_2} = 140$$

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{gbest} = 35$$

$$X_2 = [3.35 \;\; 3.2 \;\; 1.5 \;\; 6.4]$$

- Step 5: Evaluate fitness $f_2 = 3.35^2 + 3.2^2 + 1.5^2 + 6.4^2 = 64.67$

- Step 6: Update population

$$Pop = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 0 & 3 & 1 & 5 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 35 \\ 102 \\ 113 \end{bmatrix}$$

- Step 7: Update $p_{best,2}$ as the new solution is better than $p_{best,2}$

$$p_{best,2} = [3.35 \quad 3.2 \quad 1.5 \quad 6.4]$$

$$f_{pbest_2} = 64.67$$

- Step 8: No update in g_{best} as new solution is not better

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{gbest} = 35$$

Third Solution: Generation

$w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$

- Step 1: Generate two sets of random vectors

$r_1 = [0.2\ \ 0.7\ \ 0.4\ \ 0.9] \qquad r_2 = [0.9\ \ 0.2\ \ 0.1\ \ 0.4]$

$v_3 = [7\ \ 4\ \ 1\ \ 4]$

$X_3 = [0\ \ 3\ \ 1\ \ 5] \qquad f = 35$

$p_{best,3} = [0\ \ 3\ \ 1\ \ 5] \qquad f_{pbest_3} = 35$

$g_{best} = [0\ \ 3\ \ 1\ \ 5] \qquad f_{gbest} = 35$

- Step 2: Determine velocity

$$v_3 = 0.7 \times [7\ 4\ 1\ 4] + 1.5 \times [0.2\ \ 0.7\ \ 0.4\ \ 0.9] \times ([0\ 3\ 1\ 5] - [0\ 3\ 1\ 5]) + 1.5 \times [0.9\ \ 0.2\ \ 0.1\ \ 0.4] \times ([0\ 3\ 1\ 5] - [0\ 3\ 1\ 5])$$

$$v_3 = [4.9\ \ 2.8\ \ 0.7\ \ 2.8]$$

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i\right) + c_2 r_2 \left(g_{best} - X_i\right)$$

$$X_i = X_i + v_i$$

- Step 3: Determine position

$$X_3 = [0\ 3\ 1\ 5] + [4.9\ \ 2.8\ \ 0.7\ \ 2.8] = [4.9\ \ 5.8\ \ 1.7\ \ 7.8]$$

Third Solution: Updating

- Step 4: Check bounds, bound if violation

$X_3 = [4.9 \;\; 5.8 \;\; 1.7 \;\; 7.8]$ $\boxed{0 \le x_i \le 10}$

$$v_3 = [4.9 \quad 2.8 \quad 0.7 \quad 2.8]$$

$$X_3 = [0 \quad 3 \quad 1 \quad 5] \quad f_3 = 35$$

$$p_{best,3} = [0 \quad 3 \quad 1 \quad 5] \quad f_{pbest_3} = 35$$

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \quad f_{gbest} = 35$$

- Step 5: Evaluate fitness $f_3 = 4.9^2 + 5.8^2 + 1.7^2 + 7.8^2 = 121.38$

- Step 6: Update population

$$Pop = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 2 & 1 & 4 & 9 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 121.38 \\ 102 \\ 113 \end{bmatrix}$$

- Step 7: No update of $p_{best,3}$ as new solution is not better.

$$p_{best,3} = [0 \quad 3 \quad 1 \quad 5] \quad f_{pbest_3} = 35$$

- Step 8: No update in g_{best} as new solution is not better.

$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \quad f_{gbest} = 35$$

Fourth Solution: Generation

$w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$

- Step 1: Generate two sets of random vectors

$r_1 = [0.7\ \ 0.5\ \ 0.8\ \ 0.1]$ $r_2 = [0.8\ \ 0.1\ \ 0.7\ \ 0.9]$

$$v_4 = [3\ \ 0\ \ 2\ \ 1]$$
$$X_4 = [2\ \ 1\ \ 4\ \ 9] \qquad f = 102$$
$$p_{best,4} = [2\ \ 1\ \ 4\ \ 9] \qquad f_{pbest_4} = 102$$
$$g_{best} = [0\ \ 3\ \ 1\ \ 5] \qquad f_{gbest} = 35$$

- Step 2: Determine velocity

$$\begin{aligned} v_4 &= 0.7 \text{ x } [3\ \ 0\ \ 2\ \ 1] + \\ &\quad 1.5 \text{ x } [0.7\ \ 0.5\ \ 0.8\ \ 0.1] \text{ x } ([2\ 1\ 4\ 9] - [2\ 1\ 4\ 9]) + \\ &\quad 1.5 \text{ x } [0.8\ \ 0.1\ \ 0.7\ \ 0.9] \text{ x } ([0\ 3\ 1\ 5] - [2\ 1\ 4\ 9]) \\ v_4 &= [-0.3\ \ 0.3\ \ -1.75\ \ -4.7] \end{aligned}$$

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$
$$X_i = X_i + v_i$$

- Step 3: Determine position

$$\begin{aligned} X_4 &= [2\ \ 1\ \ 4\ \ 9] + [-0.3\ \ 0.3\ \ -1.75\ \ -4.7] \\ &= [1.7\ \ 1.3\ \ 2.25\ \ 4.3] \end{aligned}$$

Fourth Solution: Updating

- Step 4: Check bounds, bound if violation $\boxed{0 \le x_i \le 10}$

$$v_4 = [-0.3 \quad 0.3 \quad -1.75 \quad -4.7]$$
$$X_4 = [2 \quad 1 \quad 4 \quad 9] \qquad f = 102$$
$$p_{best,4} = [2 \quad 1 \quad 4 \quad 9] \qquad f_{pbest_4} = 102$$
$$g_{best} = [0 \quad 3 \quad 1 \quad 5] \qquad f_{gbest} = 35$$

$$X_4 = [1.7 \;\; 1.3 \;\; 2.25 \;\; 4.3]$$

- Step 5: Evaluate fitness $f_4 = 1.7^2 + 1.3^2 + 2.25^2 + 4.3^2 = 28.13$

- Step 6: Update population

$$Pop = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 6 & 2 & 8 & 3 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 121.38 \\ 28.13 \\ 113 \end{bmatrix}$$

- Step 7: Update $p_{best,4}$ as the new solution is better than $p_{best,4}$

$$p_{best,4} = [1.7 \quad 1.3 \quad 2.25 \quad 4.3]$$
$$f_{pbest4} = 28.13$$

- Step 8: Update g_{best} as the new solution is better

$$f_{pbest4}(28.13) < g_{best}(35)$$

$$g_{best} = [1.7 \quad 1.3 \quad 2.25 \quad 4.3]$$
$$f_{gbest} = 28.13$$

Fifth Solution: Generation

$w = 0.7$, $c_1 = 1.5$, $c_2 = 1.5$, $T = 10$

- Step 1: Generate two sets of random vectors

$r_1 = [0.3 \;\; 0.8 \;\; 0.2 \;\; 0.1]$ $\qquad$ $r_2 = [0.5 \;\; 0.1 \;\; 0.2 \;\; 0.7]$

$$v_5 = [1 \;\; 6 \;\; 8 \;\; 7]$$

$$X_5 = [6 \;\; 2 \;\; 8 \;\; 3] \qquad f = 113$$

$$p_{best,5} = [6 \;\; 2 \;\; 8 \;\; 3] \qquad f_{pbest_5} = 113$$

$$g_{best} = [1.7 \;\; 1.3 \;\; 2.25 \;\; 4.3] \qquad f_{gbest} = 28.13$$

- Step 2: Determine velocity

$$\begin{aligned} v_5 &= 0.7 \times [1 \;\; 6 \;\; 8 \;\; 7] + \\ &\quad 1.5 \times [0.3 \;\; 0.8 \;\; 0.2 \;\; 0.1] \times ([6 \;\; 2 \;\; 8 \;\; 3] - [6 \;\; 2 \;\; 8 \;\; 3]) + \\ &\quad 1.5 \times [0.5 \;\; 0.1 \;\; 0.2 \;\; 0.7] \times ([1.7 \;\; 1.3 \;\; 2.25 \;\; 4.3] - [6 \; 2 \; 8 \; 3]) \\ v_5 &= [-2.52 \;\; 4.09 \;\; 3.87 \;\; 6.26] \end{aligned}$$

$$v_i = w v_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$

$$X_i = X_i + v_i$$

- Step 3: Determine position

$$\begin{aligned} X_5 &= [6 \;\; 2 \;\; 8 \;\; 3] + [-2.52 \;\; 4.09 \;\; 3.87 \;\; 6.26] \\ &= [3.48 \;\; 6.09 \;\; 11.87 \;\; 9.26] \end{aligned}$$

Fifth Solution: Updating

- Step 4: Check bounds, bound if violation $\boxed{0 \le x_i \le 10}$

$$v_5 = \begin{bmatrix} -2.52 & 4.09 & 3.87 & 6.26 \end{bmatrix}$$

$$X_5 = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \quad f = 113$$

$$p_{best,5} = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \quad f_{pbest_5} = 113$$

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \quad f_{gbest} = 28.13$$

$$X_5 = [3.48 \;\; 6.09 \;\; 11.87 \;\; 9.26]$$

$$X_5 = [3.48 \;\; 6.09 \;\; 10 \;\; 9.26]$$

- Step 5: Evaluate fitness $f_5 = 3.48^2 + 6.09^2 + 10^2 + 9.26^2 = 234.95$

- Step 6: Update population

$$Pop = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 3.48 & 6.09 & 10 & 9.26 \end{bmatrix} \qquad f = \begin{bmatrix} 159.32 \\ 64.67 \\ 121.38 \\ 28.13 \\ 234.95 \end{bmatrix}$$

- Step 7: No update of $p_{best,5}$ as new solution is not better.

$$p_{best,5} = \begin{bmatrix} 6 & 2 & 8 & 3 \end{bmatrix} \quad f_{pbest_5} = 113$$

- Step 8: No update of $g_{best,5}$ as new solution is not better.

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \quad f_{gbest} = 28.13$$

Completion of first iteration

$$P=\begin{bmatrix}4&0&0&8\\3&1&9&7\\0&3&1&5\\2&1&4&9\\6&2&8&3\end{bmatrix}\quad f=\begin{bmatrix}80\\140\\35\\102\\113\end{bmatrix}\quad P_{best}=\begin{bmatrix}4&0&0&8\\3&1&9&7\\0&3&1&5\\2&1&4&9\\6&2&8&3\end{bmatrix}\quad f_{pbest}=\begin{bmatrix}80\\140\\35\\102\\113\end{bmatrix}\quad v=\begin{bmatrix}9&6&1&8\\5&1&3&0\\7&4&1&4\\3&0&2&1\\1&6&8&7\end{bmatrix}$$

$$g_{best}=\begin{bmatrix}0&3&1&5\end{bmatrix}$$
$$f_{gbest}=35$$

$$P=\begin{bmatrix}5.5&5.1&1.75&10\\3.35&3.2&1.5&6.4\\4.9&5.8&1.7&7.8\\1.7&1.3&2.25&4.3\\3.48&6.09&10&9.26\end{bmatrix}\quad f=\begin{bmatrix}159.32\\64.67\\121.38\\28.13\\235.95\end{bmatrix}\quad P_{best}=\begin{bmatrix}4&0&0&8\\3.35&3.2&1.5&6.4\\0&3&1&5\\1.7&1.3&2.25&4.3\\6&2&8&3\end{bmatrix}\quad f_{pbest}=\begin{bmatrix}80\\64.67\\35\\28.13\\113\end{bmatrix}\quad v=\begin{bmatrix}1.5&5.1&1.75&3.8\\0.35&2.2&-7.5&-0.6\\4.9&2.8&0.7&2.8\\-0.3&0.3&-1.75&-4.7\\-2.52&4.09&3.87&6.26\end{bmatrix}$$

$$g_{best}=\begin{bmatrix}1.7&1.3&2.25&4.3\end{bmatrix},\ f_{gbest}=28.13$$

Second Iteration: Generation of first solution

$w = 0.7,\ c_1 = 1.5,\ c_2 = 1.5,\ T = 10$

$$P = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 1.7 & 1.3 & 2.25 & 4.3 \\ 3.48 & 6.09 & 10 & 9.26 \end{bmatrix} \quad v = \begin{bmatrix} 1.5 & 5.1 & 1.75 & 3.8 \\ 0.35 & 2.2 & -7.5 & -0.6 \\ 4.9 & 2.8 & 0.7 & 2.8 \\ -0.3 & 0.3 & -1.75 & -4.7 \\ -2.52 & 4.09 & 3.87 & 6.26 \end{bmatrix}$$

$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix},\ f_{gbest} = 28.13$

$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_{pbest_1} = 80$

- Step 1: Generate two sets of random vectors

$$r_1 = [0.7\ \ 0.2\ \ 0.8\ \ 0.1]$$
$$r_2 = [0.9\ \ 0.3\ \ 0.2\ \ 0.5]$$

- Step 2: Determine velocity

$$v_1 = 0.7 \times [1.5\ \ 5.1\ \ 1.75\ \ 11.8] + 1.5 \times [0.7\ \ 0.2\ \ 0.8\ \ 0.1] \times ([4\ \ 0\ \ 0\ \ 8] - [5.5\ \ 5.1\ \ 1.75\ \ 10]) + 1.5 \times [0.9\ \ 0.3\ \ 0.2\ \ 0.5] \times ([1.7\ \ 1.3\ \ 2.25\ \ 4.3] - [5.5\ \ 5.1\ \ 1.75\ \ 10])$$

$$v_1 = [-5.65\ \ 0.33\ \ -0.73\ \ 3.68]$$

- Step 3: Determine position

$$X_1 = [5.5\ \ 5.1\ \ 1.75\ \ 10] + [-5.65\ \ 0.33\ \ -0.73\ \ 3.68] = [-0.15\ \ 5.43\ \ 1.02\ \ 13.68]$$

First Solution: Updating

- Step 4: Check bounds, bound if violation $\boxed{0 \le x_i \le 10}$

$$v_1 = \begin{bmatrix} -5.65 & 0.33 & -0.73 & 3.68 \end{bmatrix}$$

$$X_1 = \begin{bmatrix} 5.5 & 5.1 & 1.75 & 10 \end{bmatrix} \quad f = 159.32$$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_{pbest_1} = 80$$

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \quad f_{gbest} = 28.13$$

$$X_1 = [-0.15 \;\; 5.43 \;\; 1.02 \;\; 13.68] \quad X_1 = [0 \;\; 5.43 \;\; 1.02 \;\; 10]$$

- Step 5: Evaluate fitness $f_1 = 0^2 + 5.43^2 + 1.02^2 + 10^2 = 130.53$

- Step 6: Update the population

$$P = \begin{bmatrix} 0 & 5.43 & 1.02 & 10 \\ 3.35 & 3.2 & 1.5 & 6.4 \\ 4.9 & 5.8 & 1.7 & 7.8 \\ 3.1 & 7.6 & 2.95 & 5 \\ 2.2 & 6.35 & 10 & 10 \end{bmatrix} \qquad f = \begin{bmatrix} 130.53 \\ 64.67 \\ 121.38 \\ 101.07 \\ 245.16 \end{bmatrix}$$

- Step 7: No update of $p_{best,1}$ as new solution is not better than $p_{best,1}$

$$p_{best,1} = \begin{bmatrix} 4 & 0 & 0 & 8 \end{bmatrix} \quad f_{pbest_1} = 80$$

- Step 8: No update of g_{best} as new solution is not better.

$$g_{best} = \begin{bmatrix} 1.7 & 1.3 & 2.25 & 4.3 \end{bmatrix} \quad f_{gbest} = 28.13$$

Pseudocode

Input: Fitness function, lb, ub, N_p, T , w, c_1 and c_2

p_{best}: N_p x D, f_{pbest}: N_p x 1
g_{best}: 1 x D, f_{gbest}: 1 x 1

For T iterations
Total FE $= N_p + N_p T$

1. Initialize a random population (P) and velocity (v) within the bounds
2. Evaluate the objective function value (f) of P — FE = N_p
3. Assign p_{best} as P and f_{pbest} as f
4. Identify the solution with best fitness and assign that solution as g_{best} and fitness as f_{gbest}

for t = 1 to T
  for i = 1 to N_p
    Determine the velocity (v_i) of i^{th} particle
    Determine the new position (X_i) of i^{th} particle
    Bound X_i
    (Generation)
    Evaluate the objective function value (f_i) of i^{th} particle — FE =1
    Update the population by including X_i and f_i
    Update $p_{best,\ i}$ and f_{pbest}
    Update g_{best} and f_{gbest}
    (Memorizing)
  end
end

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i\right) + c_2 r_2 \left(g_{best} - X_i\right)$$

$$X_i = X_i + v_i$$

$$\left.\begin{aligned} p_{best,i} &= X_i \\ f_{p_{best,i}} &= f_i \end{aligned}\right\} if \;\; f_i < f_{p_{best,i}}$$

$$\left.\begin{aligned} g_{best} &= p_{best,i} \\ f_{g_{best}} &= f_{p_{best,i}} \end{aligned}\right\} if \;\; f_{p_{best,i}} < f_{g_{best}}$$

User-specified parameter: Acceleration coefficients (c_1 and c_2)

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$
$$X_i = X_i + v_i$$

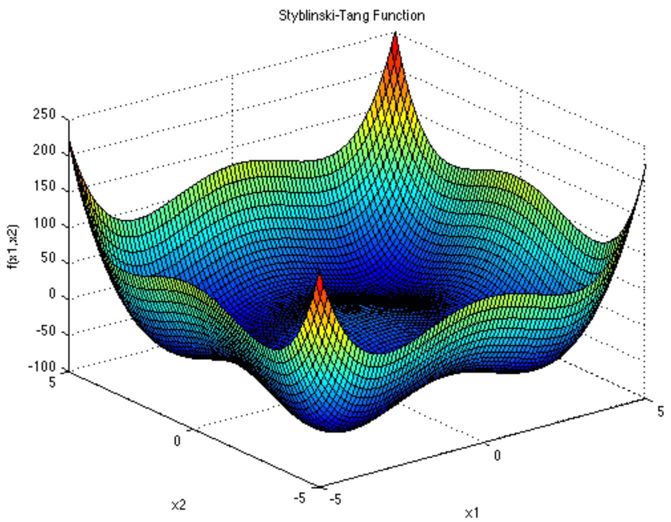
Cases	Outcome
$c_1=c_2=0$	Particles move in the same direction until it hits the search space boundary
$c_1>0$ & $c_2=0$	Particles are independent hill climbers Particles perform local search
$c_1=0$ & $c_2>0$	Entire swarm becomes one stochastic hill climber All particles get attracted to a single point
$c_1 = c_2$	Particle is attracted towards the average $p_{best,i}$ and g_{best}
$c_1>>c_2$	Particles attracted towards its $p_{best,i}$ which results in excessive wandering
$c_1<<c_2$	Particles attracted towards g_{best} and cause premature convergence towards optima
Low values of c_1 and c_2	Smooth particle trajectories
High values of c_1 and c_2	Abrupt movements

Computational Intelligence: An Introduction, Second Edition, John Wiley & Sons, 2007

Impact of c_1 and c_2

$N_p = 10, T = 50$

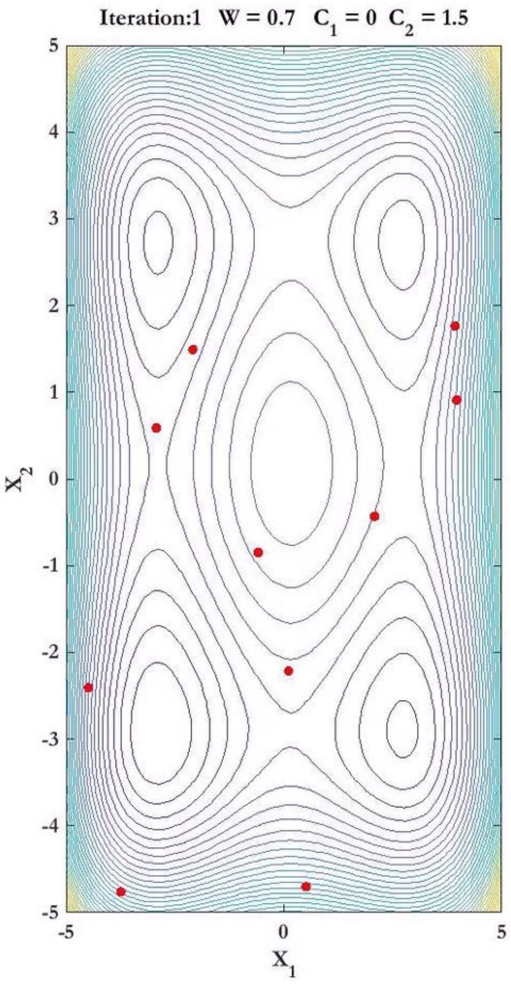
$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i\right) + c_2 r_2 \left(g_{best} - X_i\right)$$
$$X_i = X_i + v_i$$

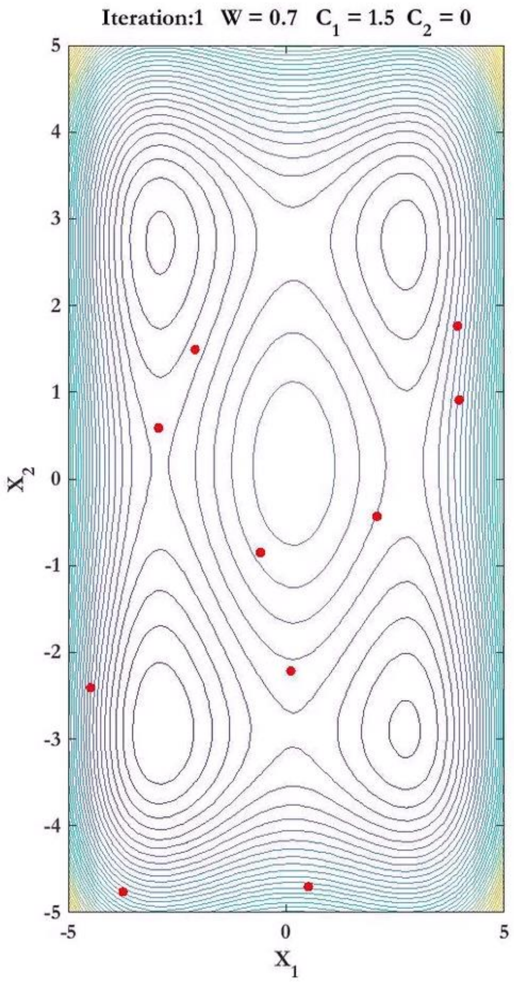


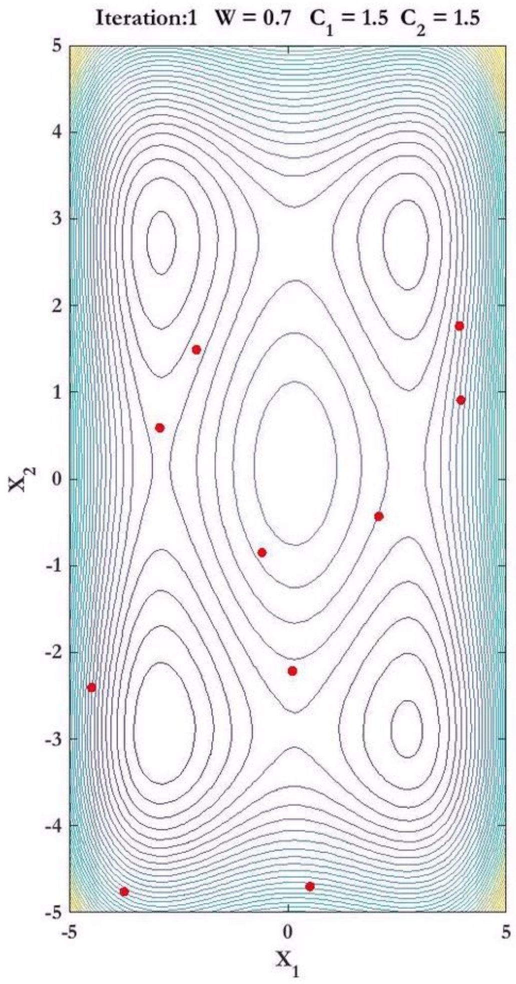
$$f = \frac{1}{2}\sum_{i=1}^{D}\left(x_i^4 - 16x_i^2 + 5x_i\right)$$
$$-5 \le x_i \le 5 \quad \forall i = 1, 2, \ldots, D$$

Global minimum

$$f^* = -39.16599$$
$$x^* = \left(-2.903534, \ldots, -2.903534\right)$$







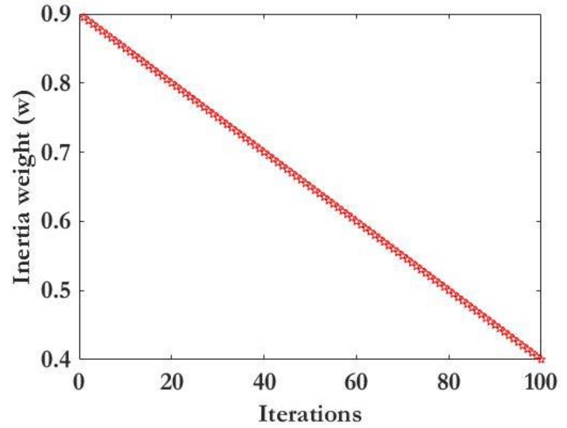
Surface plot: https://www.sfu.ca/~ssurjano/stybtang.html

User-specified parameter : Inertia weight (w)

- Control the impact of previous velocity in new direction
- Balancing exploration and exploitation
- Large inertia weight results in exploration (diverges the swarm) and small inertia causes exploitation (decelerate the particles)
- The value of w
  - can be a constant
  - multiplied with damping ratio in every iteration [a]
  - linearly decreased between w_{max} and w_{min}
    - w is frequently set as linearly decreasing from 0.9 to 0.4 [b]
  - set using constriction coefficients

$w = \alpha w$

α : *damping ratio* (user defined)



$$w = w_{max} - \frac{(w_{max} - w_{min})t}{T}$$

t : *current iteration*

T : *maximum iteration*

w_{min} *and* w_{max} *are user defined parameter*

Empirical Study of Particle Swarm Optimization, Proceedings of the 1999 Congress on Evolutionary Computation-CEC99, pp. 1945-1950 Vol. 3,1999

[a]Particle Swarm Optimization in MATLAB - Yarpiz Video Tutorial - Part 3/3, https://www.youtube.com/watch?v=ICBYrKsFPqA

[b]Particle swarm optimization: developments, applications and resources, Proceedings of the CEC 2001, pp. 81-86, 2001

Constriction coefficients

- Implemented to prevent explosion and also aid particles to converge to an optima

$$\chi = \frac{2k}{\left|2-\phi-\sqrt{\phi^2-4\phi}\right|}$$

$$0 \le k \le 1$$

$$\phi=\phi_1+\phi_2 > 4$$

Commonly used values

$$k=1,\ \phi_1 = 2.05,\ \phi_2 = 2.05$$

Constriction coefficient rule

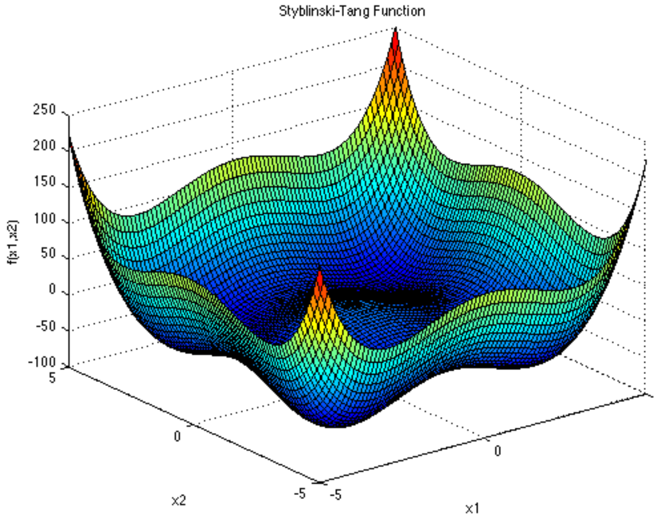
$$w=\chi \quad c_1=\chi\phi_1 \quad c_2=\chi\phi_2$$

The particle swarm - explosion, stability, and convergence in a multidimensional complex space, in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58-73. 2002
Particle Swarm Optimization in MATLAB - Yarpiz Video Tutorial - Part 3/3, https://www.youtube.com/watch?v=ICBYrKsFPqA

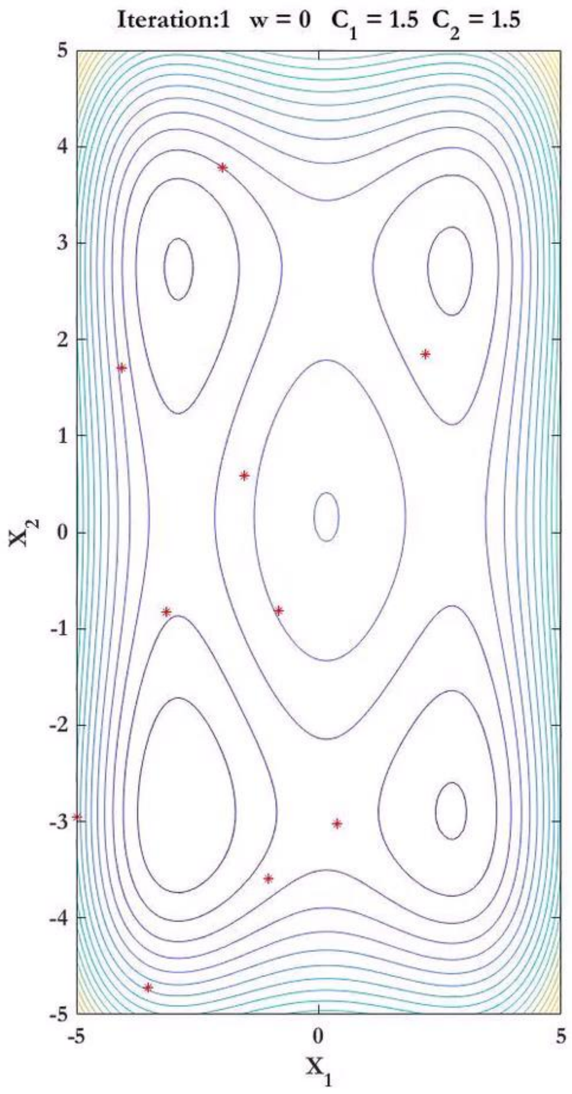
Different values of *w*

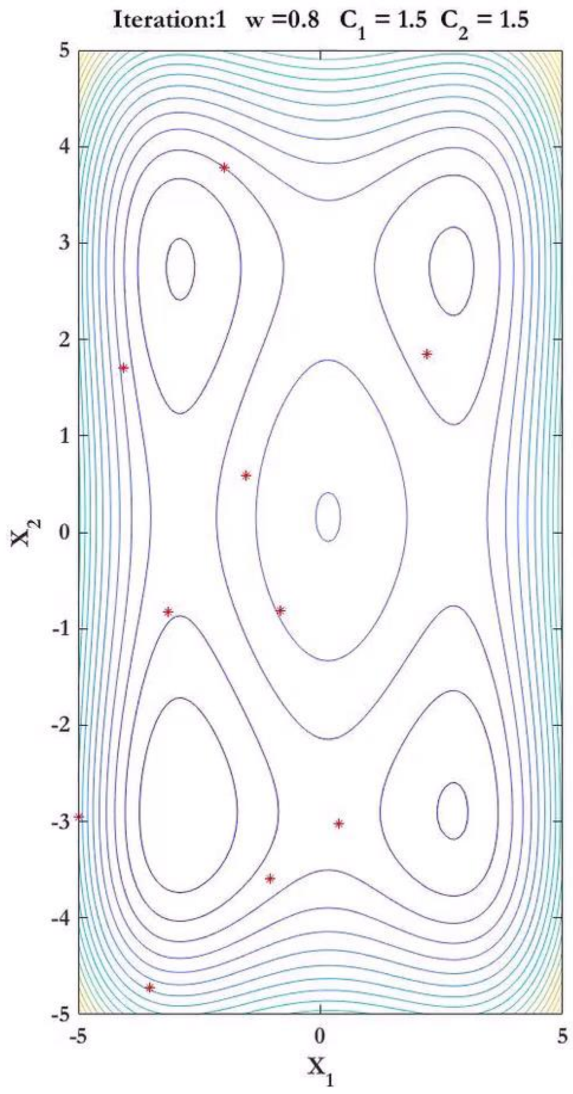
$N_p = 10, T = 50$

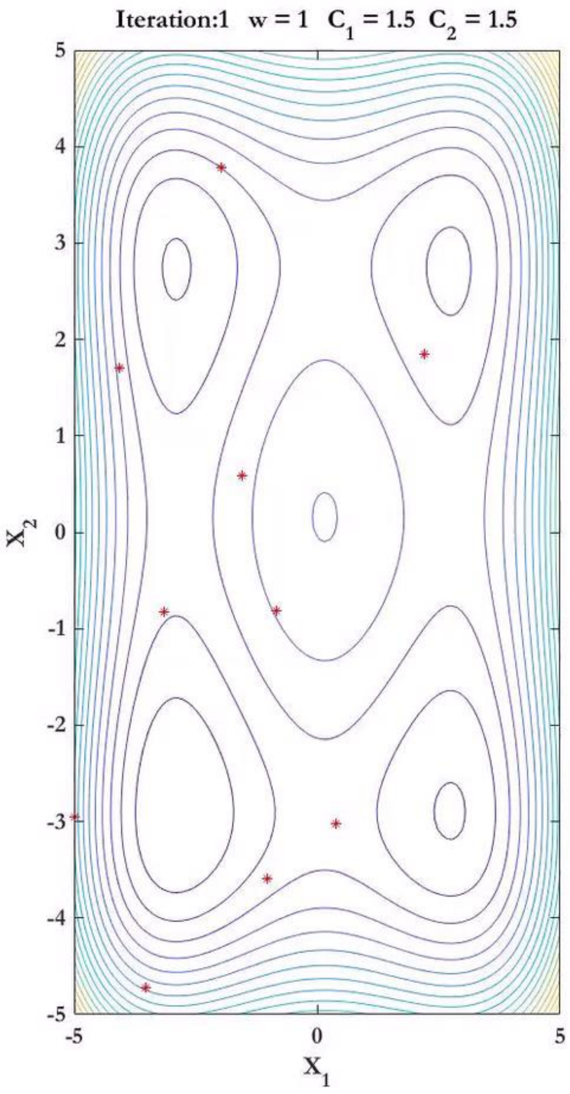
$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i\right) + c_2 r_2 \left(g_{best} - X_i\right)$$
$$X_i = X_i + v_i$$



$$f = \frac{1}{2}\sum_{i=1}^{D}\left(x_i^4 - 16x_i^2 + 5x_i\right)$$
$$-5 \le x_i \le 5 \quad \forall i = 1,2,\ldots,D$$
Global minimium
$$f^* = -39.16599$$
$$x^* = \left(-2.903534,\ldots,-2.903534\right)$$





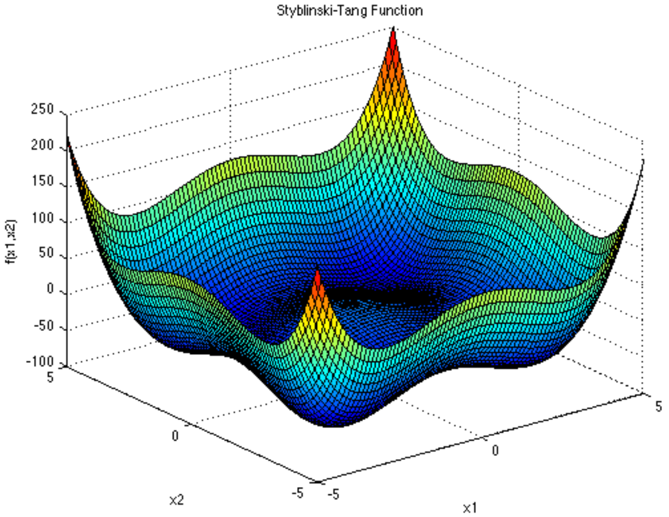


Surface plot: https://www.sfu.ca/~ssurjano/stybtang.html

Varying w

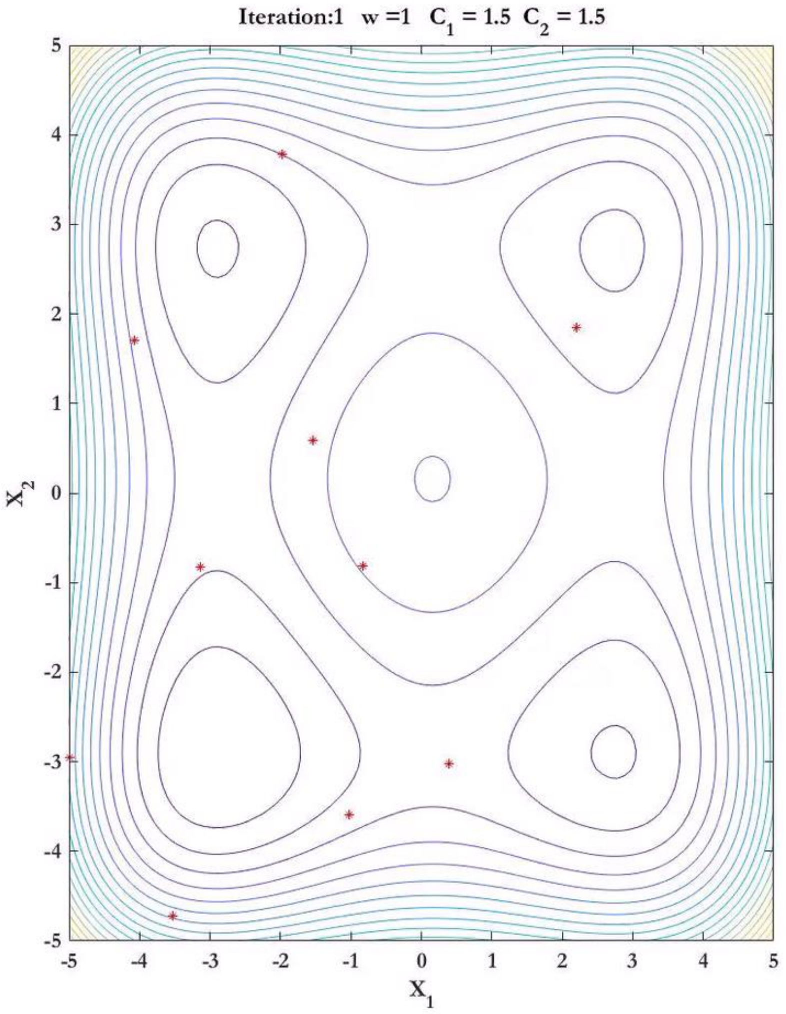
$N_p = 10$, $T = 50$
Damping ratio $= 0.99$
$w_{max} = 0.9$, $w_{min} = 0.4$

$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$
$$X_i = X_i + v_i$$

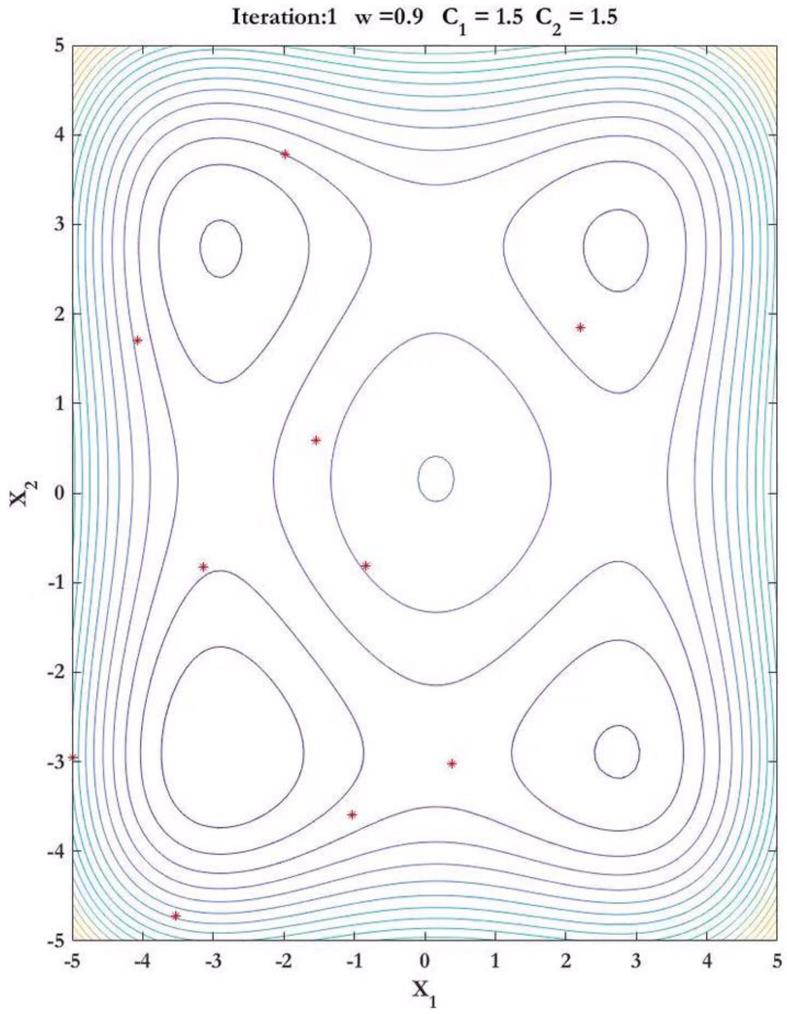


$$f = \frac{1}{2} \sum_{i=1}^{D} \left(x_i^4 - 16x_i^2 + 5x_i \right)$$
$$-5 \le x_i \le 5 \quad \forall i = 1,2,\ldots,D$$
Global minimum
$$f^* = -39.16599$$
$$x^* = \left(-2.903534, \ldots, -2.903534 \right)$$

Varied using damping ratio



Varied linearly

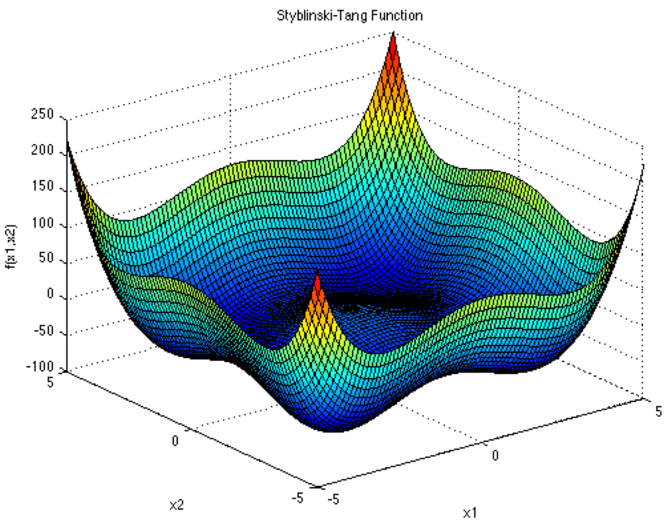


Surface plot: https://www.sfu.ca/~ssurjano/stybtang.html

Constriction coefficients

$N_p = 10$, $T = 50$
$k=1$, $\phi_1 = \phi_2 = 2.05$

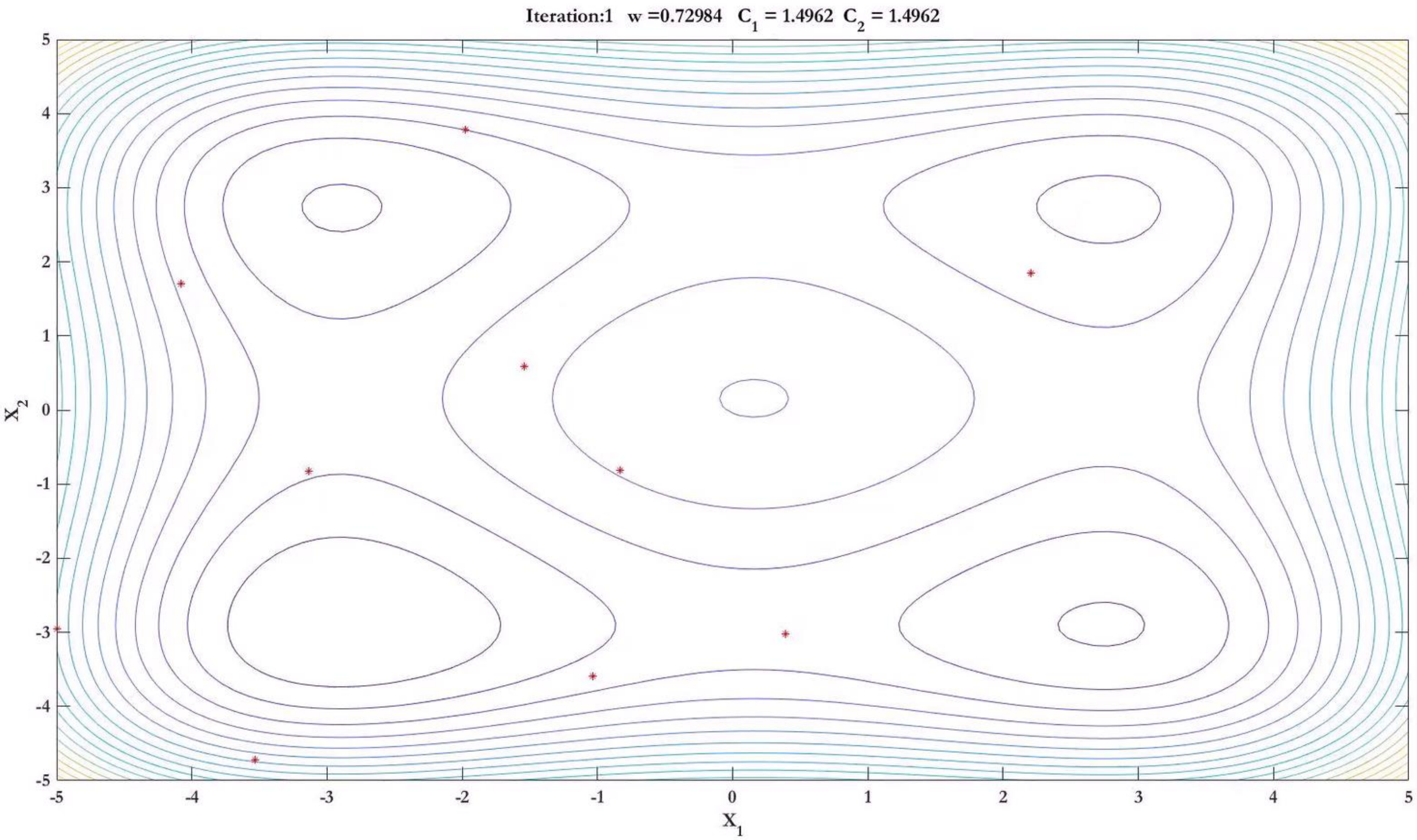
$$v_i = wv_i + c_1 r_1 \left(p_{best,i} - X_i \right) + c_2 r_2 \left(g_{best} - X_i \right)$$
$$X_i = X_i + v_i$$



$$f = \frac{1}{2}\sum_{i=1}^{D}\left(x_i^4 - 16x_i^2 + 5x_i\right)$$
$$-5 \le x_i \le 5 \quad \forall i = 1,2,\ldots,D$$

Global minimium

$$f^* = -39.16599$$
$$x^* = \left(-2.903534, \ldots, -2.903534\right)$$



Surface plot: https://www.sfu.ca/~ssurjano/stybtang.html

TLBO vs PSO

	TLBO	PSO
Phases	Teacher, Learner	No phases (Position and velocity update)
Convergence	Monotonic	Monotonic (with g_{best} and p_{best})
Parameters	Population size, termination criteria	Population size, termination criteria, inertia weight, acceleration coefficients
Generation of new solution	Only using other solutions, mean and best solution (part of population)	using velocity vector, personal best and global best (need not be the part of population)
Solution update in one iteration	Twice	Once
Selection	Greedy	New solution is always accepted (μ, λ)
Number of function evaluations	$N_p + 2N_pT$	$N_p + N_pT$

Further reading

- Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 4, 1942-1948, 1995
- The particle swarm - explosion, stability, and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation, 6(1), 58-73, 2002
- Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation, 8(3), 256-279, 2004
- A dynamic neighbourhood learning based particle swarm optimizer for global numerical optimization, Information Sciences, 209, 16-36 ,2012

Thank You !!!