Population-based Optimization:

Particle Swarm Optimization (PSO)-I

Lecture 15 – Thursday July 3, 2014

Outline

Search Methods

- Deterministic
  - Numerical Methods
  - **Graph Search Methods**
    - **Blind Search**
      - ◇ Depth-first
      - ◇ Breadth-first
      - ◇ Dijkstra
    - **Informed Search**
      - ◇ Hill climbing
      - ◇ Beam search
      - ◇ Best-first, A*
    - **Game Playing as search**
- Stochastic
  - **Trajectory Methods**
    - ◇ TS
    - ◇ SA
  - **Population-based Methods**
    - **Evolutionary Computing**
      - ◇ GA
    - **Swarm Intelligence**
      - ◇ **PSO**
      - ◇ **ACO**

Outline

- Introduction
- PSO Algorithm
- Fitness Update
- Initializations
- Neighbourhoods
- Termination Criteria
- Continuous Problems

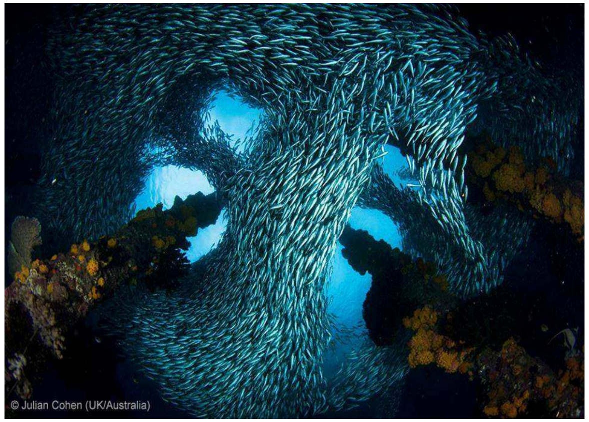
Outline

- <u>**Introduction**</u>
- PSO Algorithm
- Fitness Update
- Initializations
- Neighbourhoods
- Termination Criteria
- Continuous Problems

Introduction

- **Particle Swarm Optimization** or (PSO) is based on the behavior of a **colony or swarm of insects**, such as ants, termites, wasps and bees; a **flock of birds**; or a **school of fish**.
- The **PSO** mimics the behavior of these **social organisms**.

Bird Flock



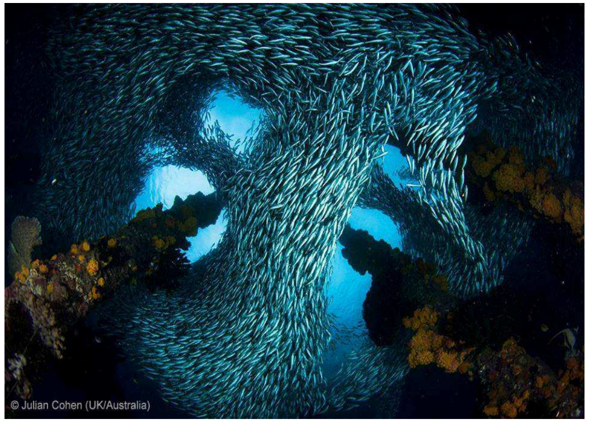
Fish School

Honey Bees Hive

Ant Foraging

Introduction

- The word ***particle*** denotes, for example, a **bee** in a colony, a **fish** in a school or a **bird** in a flock.
- Each **individual or particle** in a swarm behaves in a distributed way using its **own intelligence** and the **collective or group intelligence** of the swarm.
- As such if **one particle discovers** a good path to food, the **rest** of the swarm will also be able to **follow** the good path instantly.



Fish School

Bird Flock

Introduction

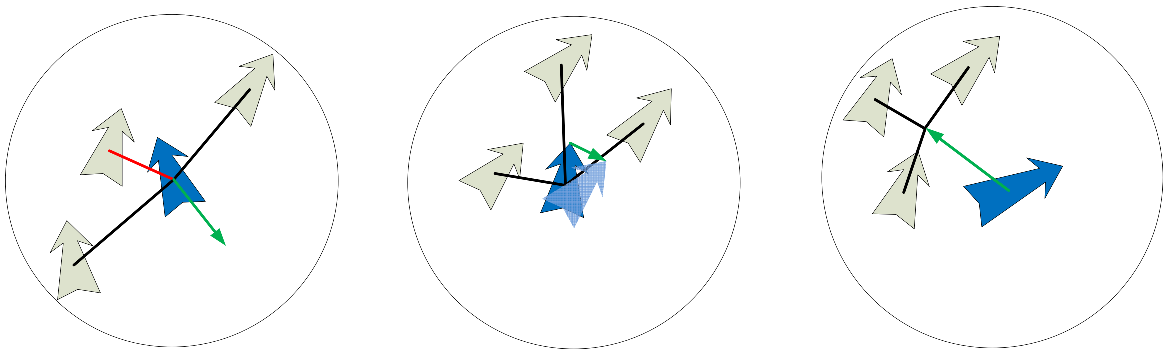
- **Bird Flocking**

Bird flocking is a behavior controlled by three simple rules:

◇ **Separation:** avoid crowding neighbors

◇ **Alignment:** steer towards average heading of neighbors

◇ **Cohesion:** steer towards average position of neighbors



Separation **Alignment** **Coherence**

Staying together but not colliding

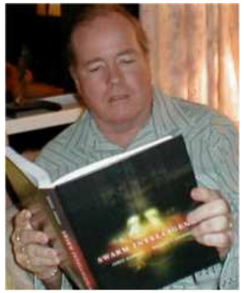
Online simulations:

http://www.red3d.com/cwr/boids/

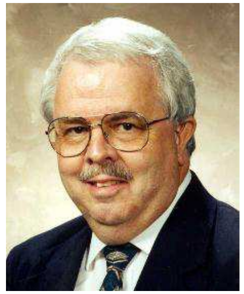
Last accessed June 28, 2014

Introduction

- Particle swarm optimization (**PSO**) is a population based **stochastic optimization** technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of **bird flocking or fish schooling**.



James Kennedy



Russell Eberhart

- Kennedy and Eberhart choose the word **particle** to mean individual or **candidate solution** (in optimization terms) as they felt it is more appropriate for the use with **velocity and acceleration**.

- These **particles** or potential solutions **fly** through the problem space by **following the current optimum particles**. This means that PSO follows simple behavior:

Emulate the success of the neighboring individuals

Introduction

- The PSO is developed based on the following model:
  - ◊ When *one bird* locates a target or food (or maximum of the objective function), it **instantaneously** transmits the information to **all other birds**.
  - ◊ All **other birds** gravitate to the **target** or food (or maximum of the objective function), but **not directly**.
  - ◊ There is a component of each **bird's own independent thinking** as well as its **past memory**.
- As such, **gradually over many iterations**, the birds go to the target (or maximum of the objective function). [1]

Introduction

- **PSO vs. GA**

Compared to **GA**, the advantages of PSO are that PSO is **easy to implement** and there are **few parameters to adjust**.

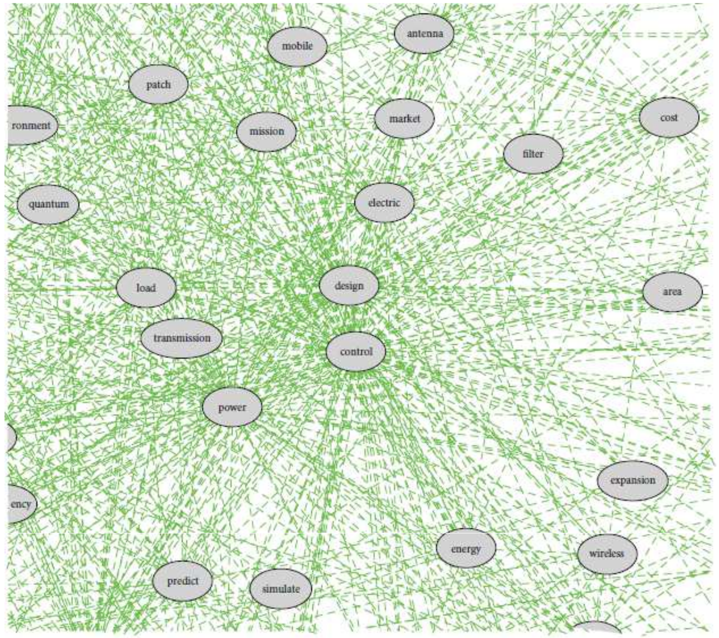
GA	PSO
Population	Swarm
Individual	Particle
Fitness	Fitness
Survival of the fittest	Less fit don’t die
Uses crossover and mutations	Uses past experience and relationship to neighbors

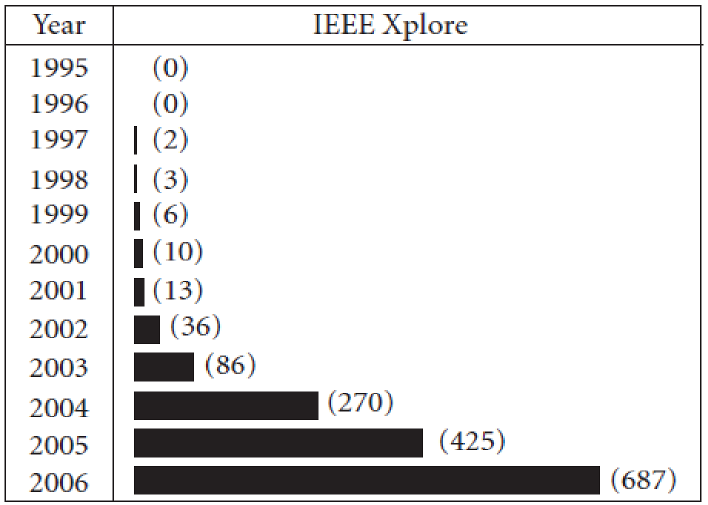
[2]

Introduction

- **Applications**
  - ◇ All areas where GA can be applied.
  - ◇ **650 different applications** are published in different domains such as antenna design, biological, medical, and pharmaceutical applications, communication networks, clustering and classification, control, distribution networks, electronics and electromagnetics, graphics and visualisation, image and video, robotics, etc.

Read PSO Applications posted on LEARN



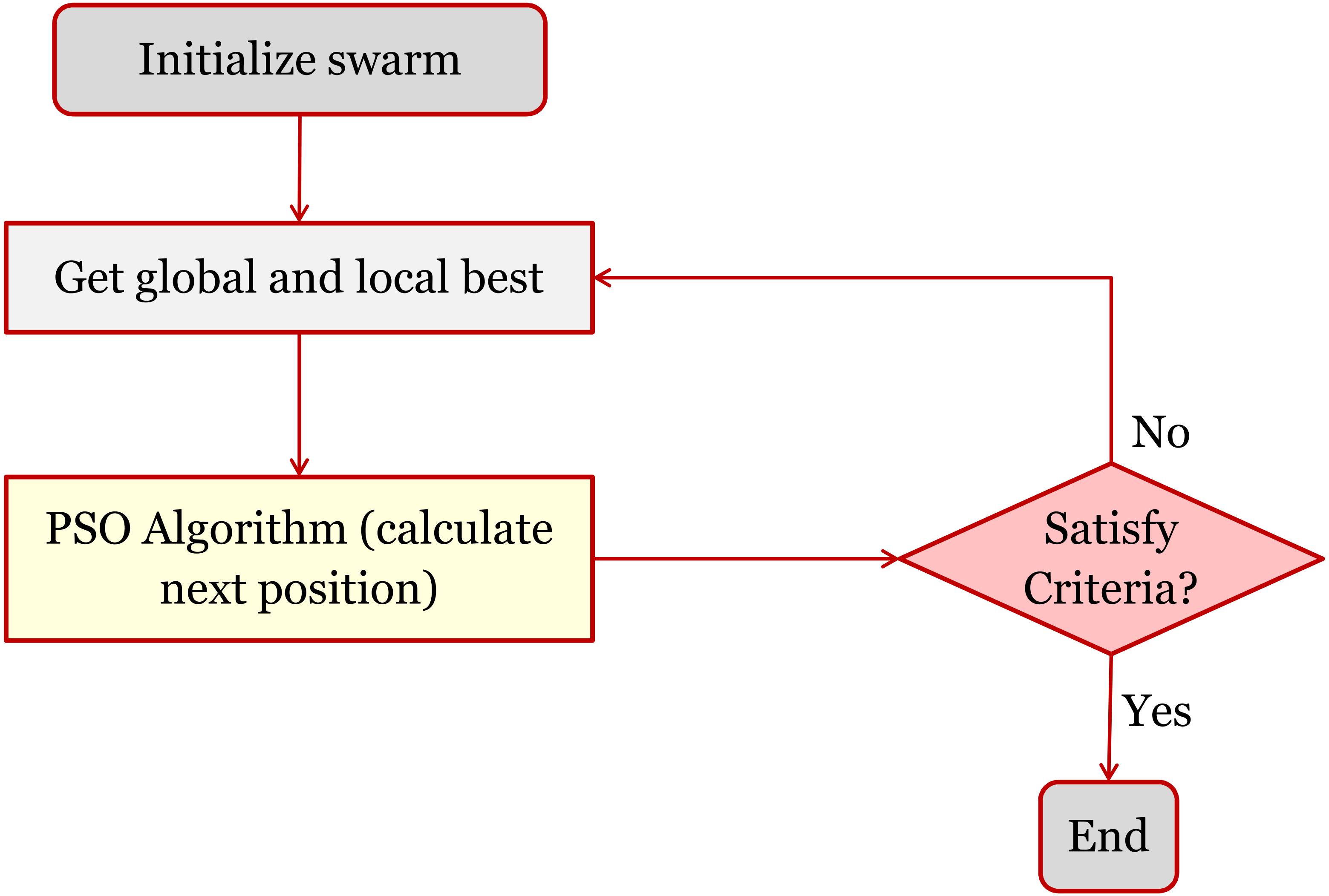


PSO papers by year [6]

Outline

- Introduction
- **PSO Algorithm**
- Fitness Update
- Initializations
- Neighbourhoods
- Termination Criteria
- Continuous Problems

PSO Algorithm



[3]

PSO Algorithm

Algorithm: PSO

Assumptions:

◇ pbest : fitness value

◇ gbest : global best

◇ lbest : local best

◇ w: inertia weight

◇ v : particle velocity

◇ x : current position (solution)

◇ rand : random number between (0,1)

◇ c1, c2 : acceleration factors

PSO Algorithm

Algorithm: PSO

For each particle

    Initialize particle

End

Do

    For each particle

        Calculate fitness value

        If the fitness value is better that the best fitness (pbest) in history.

        Set current value as the new pbest

    End

PSO Algorithm

Algorithm: PSO (cont'd)

Choose the particle with best fitness value of all the particles as the **gbest**

For each particle

v[t+1]=w*v[t]+c1*rand()*(pbest[]-x[t])+c2*rand()*(gbest[]-x[t])

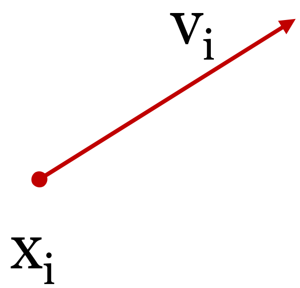
x[t+1]=x[t]+v[t+1]

End

While maximum iterations or minimum error criteria is not attained

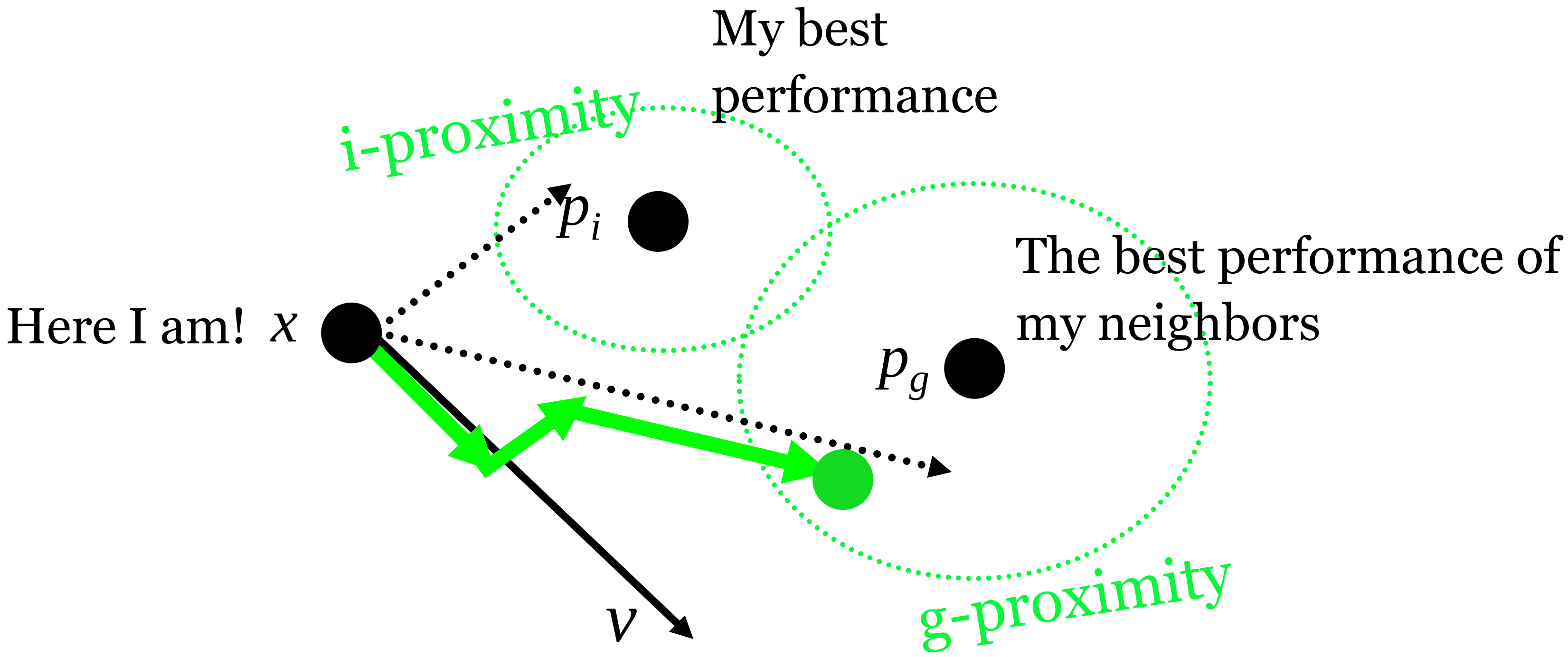
PSO Algorithm

- **Motion**
  - ◇ Each particle holds:
    - Its current position x_i ,
    - Its current velocity v_i,
    - The best position it achieved so far, **personal best**, $\boldsymbol{pbest_i}$, or sometimes $\boldsymbol{p_i}$ for short,
    - The best position achieved by particles in its neighborhood **Nbest**,
    - If the neighborhood is the whole swarm, the best achieved by the whole swarm is called **global best,** $\boldsymbol{gbest_i}$, or sometimes $\boldsymbol{p_g}$ for short.
    - If the neighborhood is restricted to few particles, the best is called **local best**, ***lbest*** or ***l***.



PSO Algorithm

- **Motion**
  - ◇ Each particle adjusts its velocity to move towards its personal best and the swarm neighborhood best,



  - ◇ After the velocity is updated, the particle adjusts its position.

PSO Algorithm

- **Motion Equations**
  - ◇ **Particle's velocity Update:**

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)$$

  - ◇ **Particle's Position Update:**

$$x_{t+1}^{id} = x_t^{id} + v_{t+1}^{id}$$

where

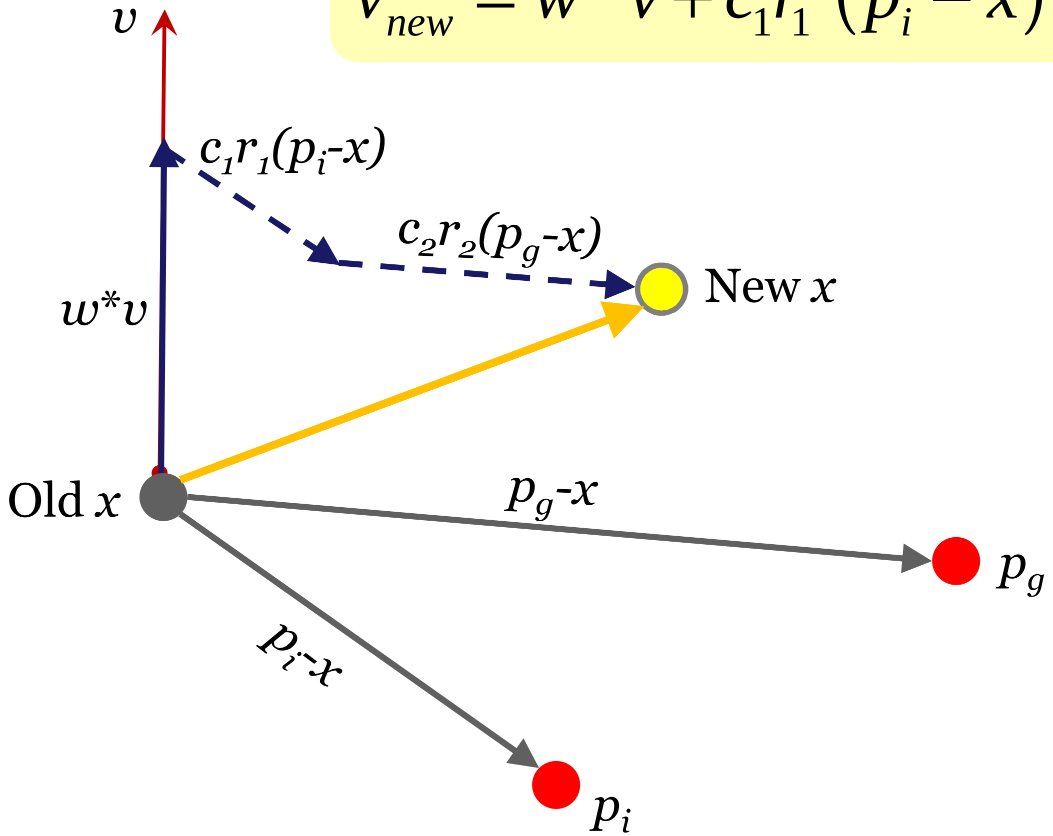
- w is the inertia weight and
- c_1, c_2 are the acceleration coefficients,
- r_1,r_2 are randomly generated numbers in [0, 1] and generated for each dimension and not for each particle,
- t is the iteration number,
- i and d are the particle number and the dimension.

PSO Algorithm

- **Motion Equations**

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id} \left(pbest_t^{id} - x_t^{id} \right) + c_2 r_2^{id} \left(Nbest_t^{id} - x_t^{id} \right)$$

$$v_{new} = w * v + c_1 r_1 \left(p_i - x \right) + c_2 r_2 \left(p_g - x \right)$$



PSO Algorithm

- **Motion Equations**
  - ◇ **Particle's velocity Update:**

$$v_{t+1}^{id} = \underbrace{w * v_t^{id}}_{\text{Inertia}} + \underbrace{c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right)}_{\text{Cognitive Component}} + \underbrace{c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)}_{\text{Social Component}}$$

    - The inertia component accommodates the fact that a bird (particle) **cannot suddenly change** its direction of movement,
    - The c1 and c2 factors balance the weights in which each particle:
      - o Trusts its **own experience**, **cognitive** component,
      - o Trusts the **swarm experience**, **social** component.

Outline

- Introduction
- PSO Algorithm
- **Fitness Update**
- Initializations
- Neighbourhoods
- Termination Criteria
- Continuous Problems

Fitness Update

- **Personal Best Update:**

  After moving, each particle updates its own personal best (assuming a minimization problem):

$$pbest_{t+1}^{i} = \begin{cases} x_{t+1}^{i} & \text{if } f(x_{t+1}^{i}) \leq f(pbest_{t}^{i}) \\ pbest_{t}^{i} & \text{otherwise} \end{cases}$$

- **Global Best Update:**

  After that, each swarm updates its global best (assuming a minimization problem):

$$Nbest_{t+1}^{i} = \arg \min_{pbest_{t+1}^{i} \in N} f(pbest_{t+1}^{i})$$

Fitness Update

PSO

Synchronous

Initialize the swarm,

While termination criteria is not met

For each particle

Update the particle's velocity,

Update the particle's position,

Update the particle's personal best,

end for

Update the Nbest,

end while

Asynchronous

Initialize the swarm,

While termination criteria is not met

For each particle

Update the particle's velocity,

Update the particle's position,

Update the particle's personal best,

Update the Nbest,

end for

end while

The neighborhood best update is moved into the particles update loop

Fitness Update

- **Synchronous version**, if the neighbourhood best is updated after all the population has been updated as well,
- **Asynchronous version**, if the neighbourhood best is updated after every particle,
- **Asynchronous** version usually produces **better results** as it causes the particles to use a **more up-to-date information**.

Outline

- Introduction
- PSO Algorithm
- Fitness Update
- **Initialization**
- Neighbourhoods
- Termination Criteria
- Continuous Problems

Initialization

- For each particle need to **initialize the particle position and velocity**.
- **Particles positions** can be **initialized randomly** in the range.
- **Particles velocities** can be **initialized to zero or small values**, large values will result in large updates which may lead to divergence.
- **Personal best** position is initialized to the **particle's initial position**.

Initialization

- **Parameters**

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id} \left(pbest_t^{id} - x_t^{id} \right) + c_2 r_2^{id} \left(Nbest_t^{id} - x_t^{id} \right)$$

◇ Using c_1=0 reduces the **velocity model** to **Social-only model** or **selfless model** (particles are all attracted to Nbest)

◇ Using c_2=0 reduces to **cognition-only model** (particles are independent hill climbers).

◇ In most applications $\mathbf{c_1 = c_2 = 1.4944}$

◇ Usually $\mathbf{c_1 + c_2 \leq 4}$. No good reason other than empiricism.

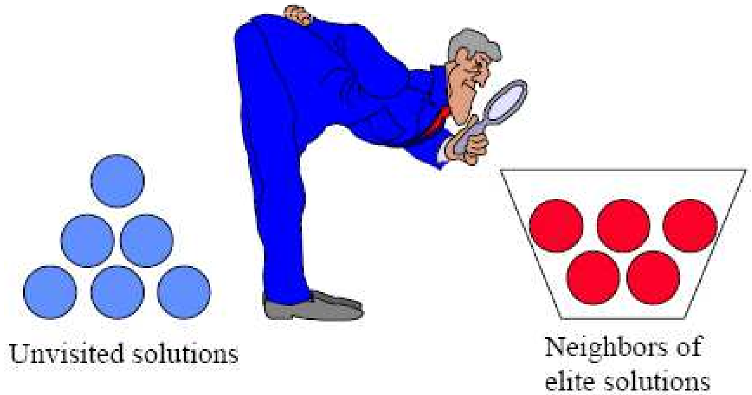
Initialization

- **Parameters**

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)$$

◇ The value of ***w*** is important to balance **exploration and exploitation**. Widely used value is 0.792.

◇ Large values promote exploration and small promote exploitation (allowing more control to cognitive and social components).



Exploration

Large *w*

Exploitation

Small *w*

Initialization

- **Parameters**

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)$$

◇ **Swarm size:** the typical range is **20 - 40**.

10 particles is large enough for most problems.

100-200 particles for some difficult or special problems.

◇ **Neighborhood size**

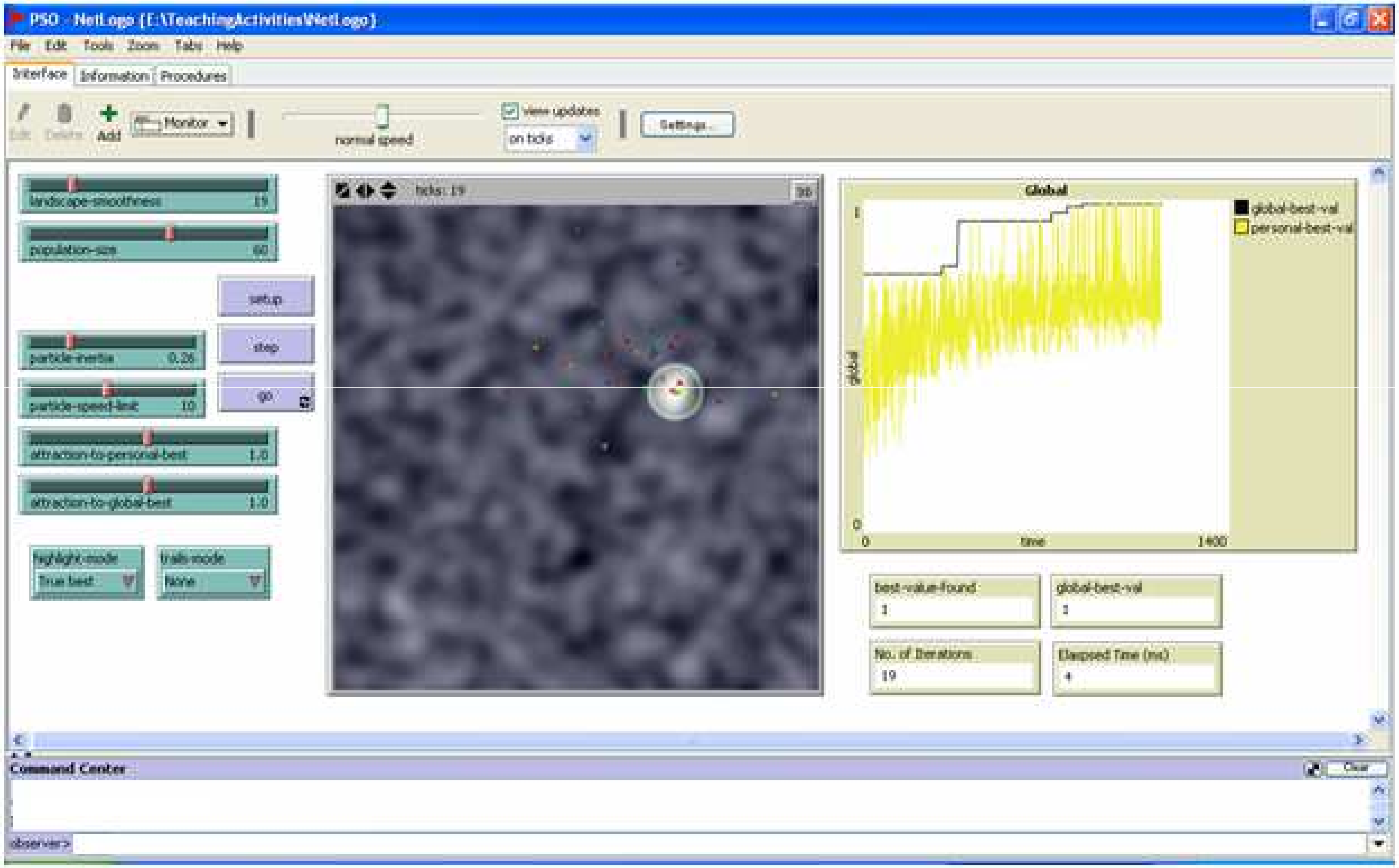
(10—50) are reported as usually sufficient.

◇ **Maximum velocity**

too low, too slow;
too high, too unstable.

Initialization

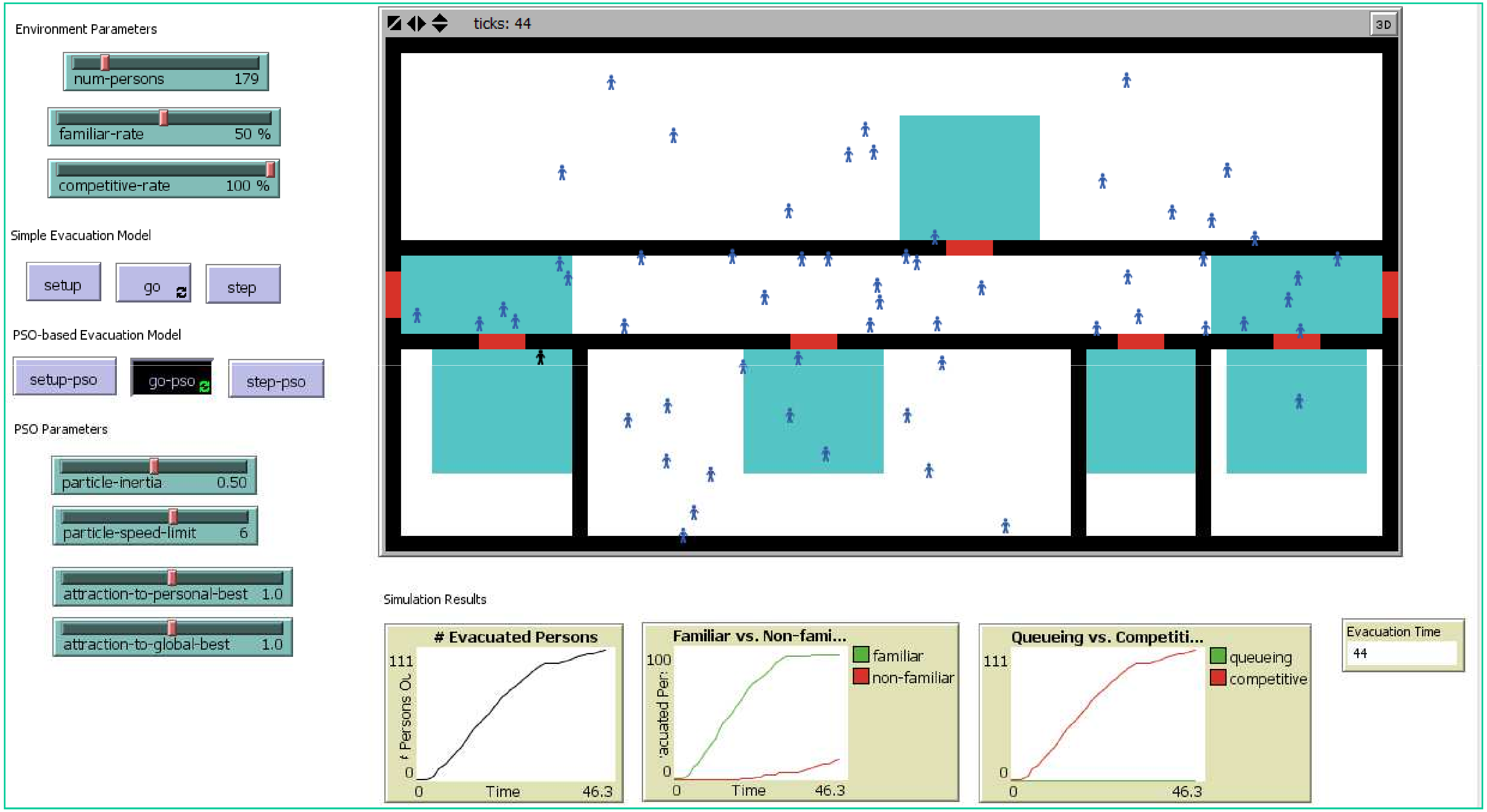
- **Demo**



http://ccl.northwestern.edu/netlogo/Last Accessed: June 28, 2014

Initialization

- **Egress – PSO Evacuation Model**



Available on LEARN

Outline

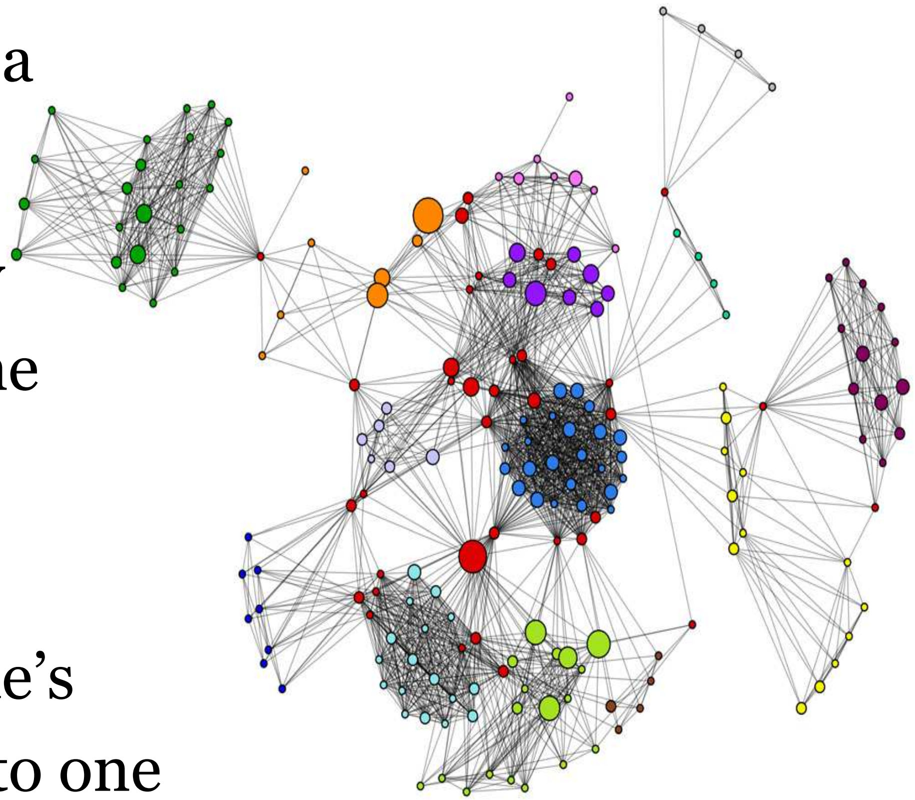
- Introduction
- PSO Algorithm
- Fitness Update
- Initialization
- **<u>Neighbourhoods</u>**
- Termination Criteria
- Continuous Problems

Neighbourhoods

- Within the PSO, particles in the **same neighborhood communicate** with one another by **exchanging information** about the success of each particle in that neighborhood.
- All particles then move towards some quantification of what is believed to be a **better position**.
- The performance of the PSO depends strongly on the **structure of the social network**.
- Selecting a **proper neighbourhood** affects the **convergence** and also helps in **avoiding getting stuck at local minima**.

Neighbourhoods

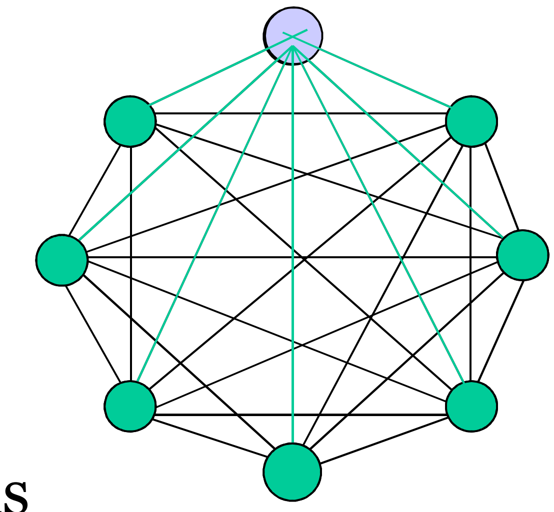
- The flow of information through a social network, depends on:



1. The **degree of connectivity** among nodes (members) of the network,
2. The **amount of clustering** (clustering occurs when a node's neighbors are also neighbors to one another), and
3. The **average shortest distance** from one node to another.

Neighbourhoods

- **Star Social Structure or gbest PSO**

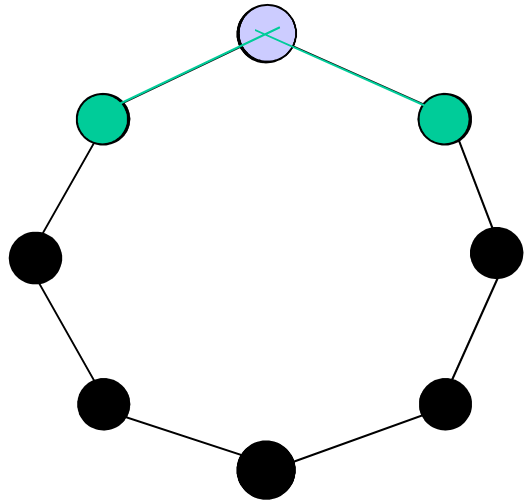


  - ◇ All particles **are interconnected**.
  - ◇ Each particle can therefore **communicate** with every other particle.
  - ◇ In this case each particle is attracted towards the best solution found by the entire swarm.
  - ◇ This leads to **gbest PSO**.
  - ◇ The gbest PSO has been shown to **converge faster** than other network structures, but with a **susceptibility to be trapped in local minima**.
  - ◇ The **gbest PSO** performs best for **unimodal problems**.

[4]

Neighbourhoods

- **Ring Topology or lbest PSO**

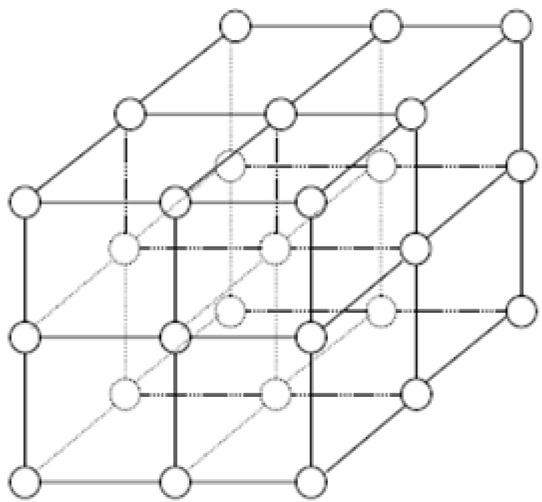


  - ◊ A particle communicates with its **immediately adjacent neighbors**. Each particle attempts to imitate its best neighbor by moving closer to the best solution found within the neighborhood.
  - ◊ The resulting PSO algorithm is generally referred to as the **lbest PSO**.
  - ◊ **Convergence is slower**, but **larger parts of the search space are covered** compared to the star structure.
  - ◊ This behavior allows the ring structure to provide better performance in terms of the quality of solutions found for **multi-modal problems** than the star structure.

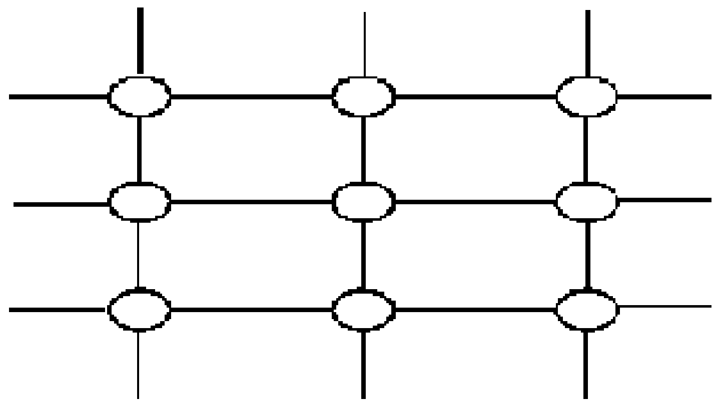
[4]

Neighbourhoods

- **The Von Neumann model**
  - ◊ The most successful neighbourhood structure was the square topology (Von Neumann model),
  - ◊ Formed by arranging the particles in a grid and connecting the neighbours above , below and to the right and left.



The complete population

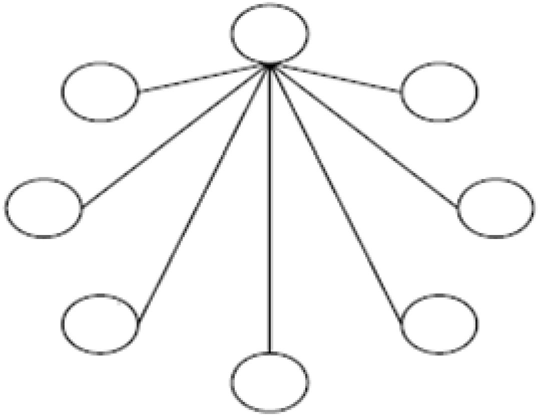


A local region

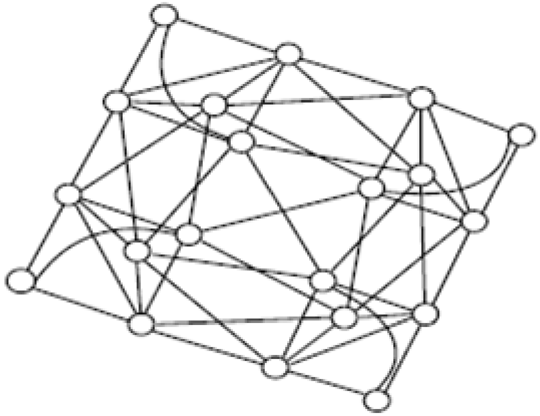
[5]

Neighbourhoods

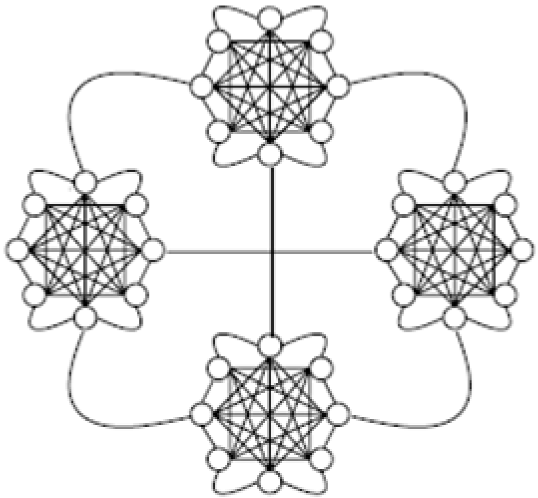
- **Others**



Wheel



Pyramid



Four Clusters

[4]

Outline

- Introduction
- PSO Algorithm
- Fitness Update
- Initialization
- Neighbourhoods
- **Termination Criteria**
- Continuous Problems

Termination Criteria

- Termination Criteria can be:
  - ◊ Max number of **iterations**
  - ◊ Max number of **function evaluations**
  - ◊ **Acceptable solution** has been found
  - ◊ **No improvement** over a number of iterations.

Outline

- Introduction
- PSO Algorithm
- Fitness Update
- Initialization
- Neighbourhoods
- Termination Criteria
- **Continuous Problems**

Continuous Problems

- Find the maximum of the function

$$f(x) = -x^2 + 2x + 11$$

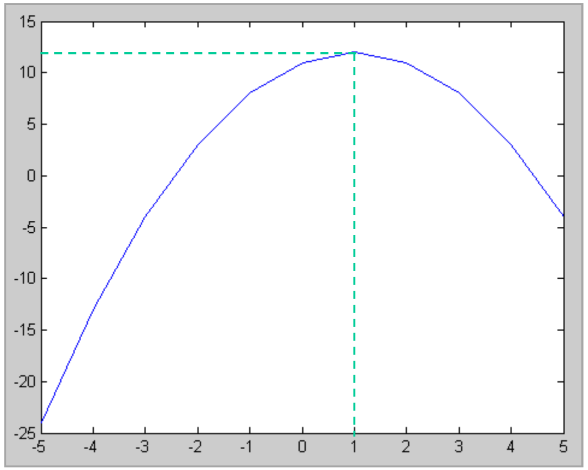
in the range **-2≤x≤2** using PSO method.

Use 4 particles (N=4) with the initial positions:

x_1=-1.5, x_2=0.0, x_3=0.5, and x_4=1.25.

Show the detailed computations for iterations 1 and 2.

$$f(x) = -x^2 + 2x + 11$$



$f_{\max}(x) = 12\big|_{\text{at } x=1}$

Continuous Problems

- **Solution:**

1. Choose the number of particles N as 4
2. The initial population, chosen randomly (given as data in this case study), can be represented as:

Particle i	Objective function $f_0^i(x)$
$x_1(0)=-1.5$	5.75
$x_2(0)=0.0$	11.0
$x_3(0)=0.5$	11.75
$x_4(0)=1.25$	11.9375

$$f(x) = -x^2 + 2x + 11$$

Continuous Problems

- **Solution:**

3. Set the initial velocities of each particle to zero

x_0^i	$f_0^i(x)$	Velocity v_0^i
-1.5	5.75	0
0.0	11.0	0
0.5	11.75	0
1.25	11.9375	0

Set the iteration number as *t*=1 and go to step 4.

Continuous Problems

- **Solution:**

4. a. Find personal best and global best

x_0^i	$f_0^i(x)$	$pbest_0^i$
-1.5	5.75	-1.5
0.0	11.0	0.0
0.5	11.75	0.5
1.25	11.9375	1.25

$gbest_0 = 1.25$

Continuous Problems

- **Solution:**

4. b. Find the velocities as (by assuming $c_1=c_2=1$ and using the random numbers in the range [0,1] as $r_1=0.3294$ and $r_2=0.9542$):

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(gbest_t^{id} - x_t^{id}\right)$$

$$v_1^i = 0 + 0.3294\left(pbest_0^i - x_0^i\right) + 0.9542\left(gbest_0^i - x_0^i\right)$$

x_0^i	v_0^i	$pbest_0^i$	v_1^i
-1.5	0	-1.5	2.6241
0.0	0	0.0	1.1927
0.5	0	0.5	0.7156
1.25	0	1.25	0.0

$gbest_0 = 1.25$

Continuous Problems

- **Solution:**

4. c. Find the new values of particles' positions using the following equation:

$$x_{t+1}^{id} = x_t^{id} + v_{t+1}^{id} \quad \text{or} \quad x_1^i = x_0^i + v_1^i$$

x_0^i	v_1^i	x_1^i
-1.5	2.6241	1.1241
0.0	1.1927	1.1927
0.5	0.7156	1.2156
1.25	0.0	1.25

Continuous Problems

- **Solution:**

5. Evaluate the objective function values at the current positions.

x_1^i	$f_1^i(x)$
1.1241	11.9846
1.1927	11.9629
1.2156	11.9535
1.25	11.9375

$$f(x) = -x^2 + 2x + 11$$

Check the **convergence** of the current solution. Since the values of the new position did not converge, we increment the iteration number as ***t*=2** and go to step 4.

Continuous Problems

- **Solution:**

4. a. Find personal best and global best

x_1^i	$f_1^i(x)$	$pbest_1^i$
1.1241	11.9846	1.1241
1.1927	11.9629	1.1927
1.2156	11.9535	1.2156
1.25	11.9375	1.25

$gbest_1 = 1.1241$

Continuous Problems

- **Solution:**

4. b. Compute the new velocities as (by assuming w=1, c_1=c_2=1 and using the random numbers in the range [0,1] as r_1=0.1482 and r_2=0.4867):

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(gbest_t^{id} - x_t^{id}\right)$$

$$v_2^i = v_1^i + 0.1482\left(pbest_1^i - x_1^i\right) + 0.4867\left(gbest_1^i - x_1^i\right)$$

x_1^i	v_1^i	$pbest_1^i$	v_2^i
1.1241	2.6241	1.1241	2.624
1.1927	1.1927	1.1927	1.1593
1.2156	0.7156	1.2156	0.6711
1.25	0.0	1.25	-0.0613

$$gbest_1 = 1.1241$$

Continuous Problems

- **Solution:**

4. c. Computer the new values of particles' positions using the following equation:

$$x_{t+1}^{id} = x_t^{id} + v_{t+1}^{id} \qquad \text{or} \qquad x_2^i = x_1^i + v_2^i$$

x_1^i	v_2^i	x_2^i
1.1241	2.624	3.7481
1.1927	1.1593	2.3520
1.2156	0.6711	1.8867
1.25	-0.0613	1.1887

Continuous Problems

- **Solution:**

6. Find the objective function values at the current positions.

x_2^i	$f_2^i(x)$
3.7481	4.4480
2.3520	10.1721
1.8867	11.2138
1.1887	11.9644

$$f(x) = -x^2 + 2x + 11$$

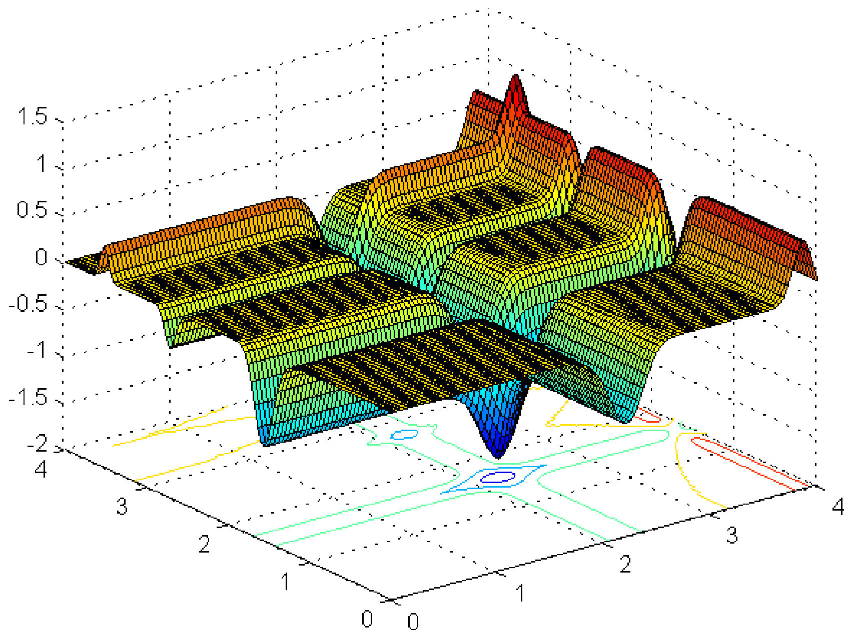
Check the **convergence** of the current solution. Since the values of the new positions did not converge, we increment the iteration number as ***t*=3** and go to step 4 until the convergence of the process is achieved.

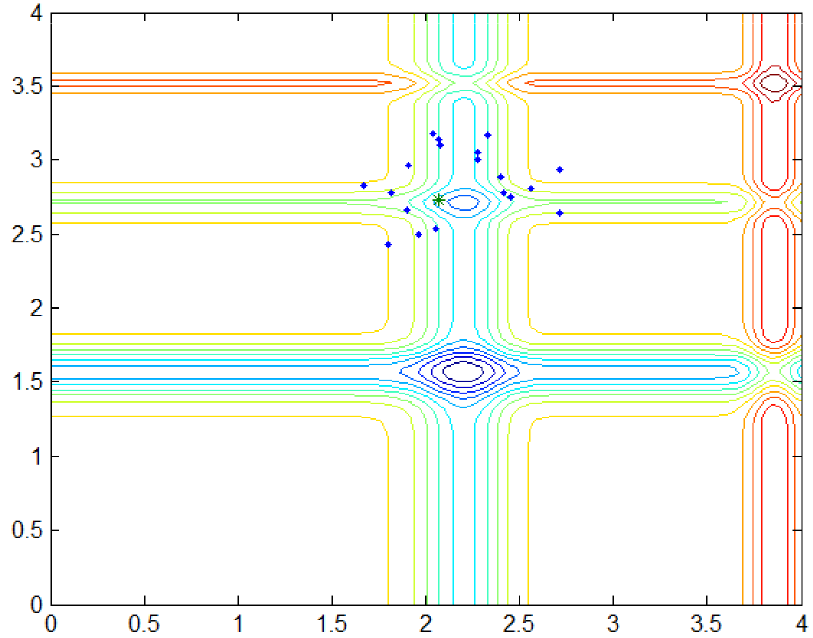
Continuous Problems

- **Assignment-4:**

The given Matlab code implements PSO to find the global minimum of Michaelwicz's 2D function:

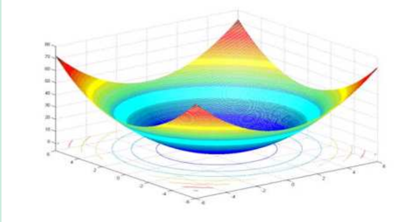
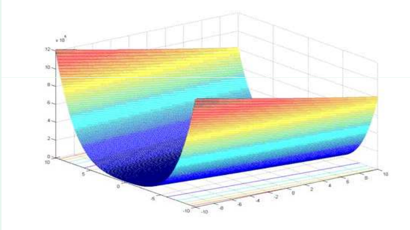
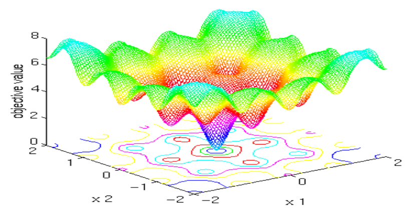
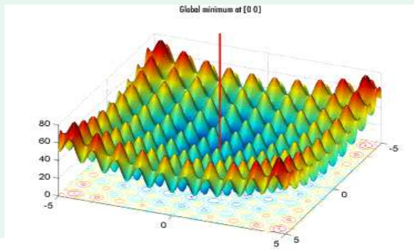
$$f(x,y) = -\sin(x)\sin^{20}\left(\frac{x^2}{\pi}\right) - \sin(y)\sin^{20}\left(\frac{2y^2}{\pi}\right)$$





Continuous Problems

- **PSO vs. GA**

Function	Formula	Plot
Spherical	$f(x)=\sum_{i=1}^{d} x_i^2$	
Rosenbrock	$f(x)=\sum_{i=1}^{d-1}\left[(1-x_i)^2+100\left(x_{i+1}-x_i^2\right)^2\right]$	
Ackley	$f(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+e$	
Rastrigin	$f(x)=\sum_{i=1}^{d}x_i^2-10\cos(2\pi x_i)+10$	

Continuous Problems

- **PSO vs. GA**

A comparison is made between **PSO** and **GAs** using the four functions (Spherical, Rosenbrock, Ackley and Rastrigin):

◇ Both algorithms use **10 particles (individuals)** and run for 1000 iterations, using Clerc and Kennedy parameters.

◇ For a dimensionality of 10,

◇ The results are the **averages** reported over **20 runs**.

[2]

Continuous Problems

- **PSO vs. GA**

Benchmark	**GA - Elitism**	**PSO - *gbest***
Spherical	0.0099	**$2.3273e^{-17}$**
Rosenbrock	**5.5760**	9.6467
Ackley	0.9451	**0.3047**
Rastrigin	38.2186	**18.7730**

[2]

References

1. Singiresu S. Rao. *Engineering Optimization: Theory and Practice*. 4th Edition,Wiley, 2009.

2. M. Kamel. ECE457A: Cooperative and Adaptive Algorithms. University of Waterloo, 2010-2011.

3. Said M. Mikki and Ahmed A. Kishk. *Particle Swarm Optimizaton: A Physics-Based Approach*. Morgan & Claypool, 2008.

4. Andries P. Engelbrecht. *Computational Intelligence: An Introduction*. 2nd Edition, John Wiley & Sons Ltd, 2007.

5. J. Kennedy and R. Mendes. “Neighbourhood Topologies in Fully Informed and best-of-neighbourhood Particle Swarms”.
IEEE Transactions on Systems, Man and Cybernetics – Part C.
Vol. 36, Issue 4, pp. 515-519, 2006.

6. Poli, R. (2008). "Analysis of the publications on the applications of particle swarm optimisation". Journal of Artificial Evolution and Applications 2008: 1–10. doi:10.1155/2008/685175.