Particle swarm Optimization and its Applications

By

Dr. B. K. Panigrahi

Professor

Department of Electrical Engineering

IIT Delhi, New Delhi-110016

OUTLINE

- ***Introduction***
- Paticle Swarm Optimization

General Optimization Algorithms

Each possible solution is evaluated

- Deterministic algorithms

- Stochastic algorithms

Deterministic Algorithms

- Hill climbing
- Branch and bound
- Depth-first
- Breadth-first
- Best-first
- lambda iteration
- Newton's method
- Gradient method
- Linear Programming
- Interior Point method

Stochastic Algorithms

- Genetic Algorithm
- Simulated Annealing
- Evolutionary Programming
- Particle Swarm Optimization
- Bacterial Foraging
- Clonal Algorithm
- Memetic Algorithm
- Ant Colony Optimization
- Frog Leaping Algorithm

Why EC ????

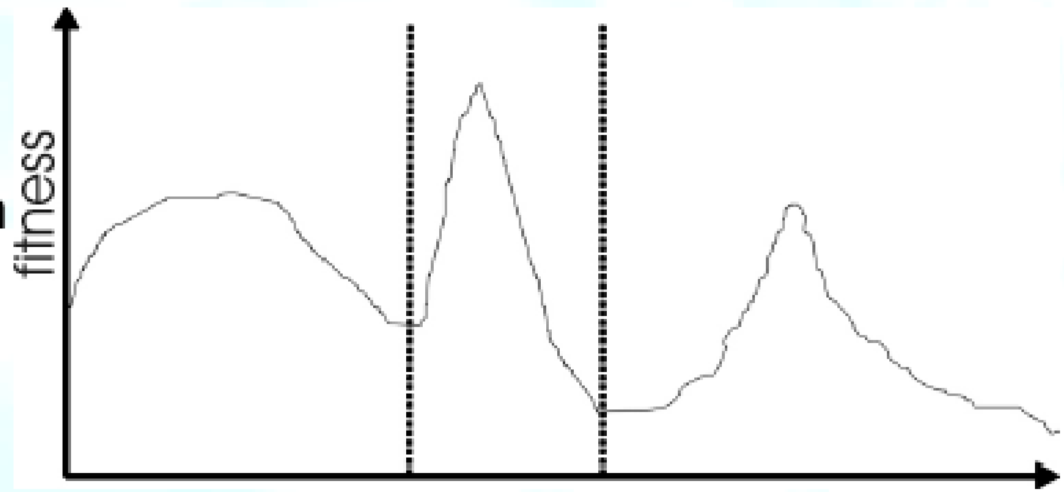
1. MULTIMODAL PROBLEMS AND MULTIPLE SOLUTIOSN

- MULTIMODAL PROBLEMS
- EVOLUTION IN MULTIMODAL PROBLEMS
- NICHING GENETIC ALGORITHMS

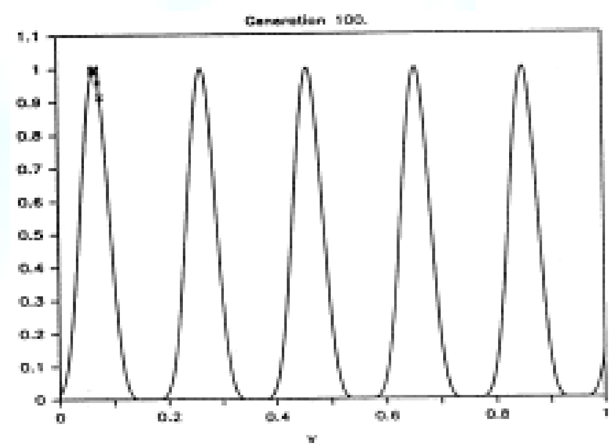
Guess !!!!!

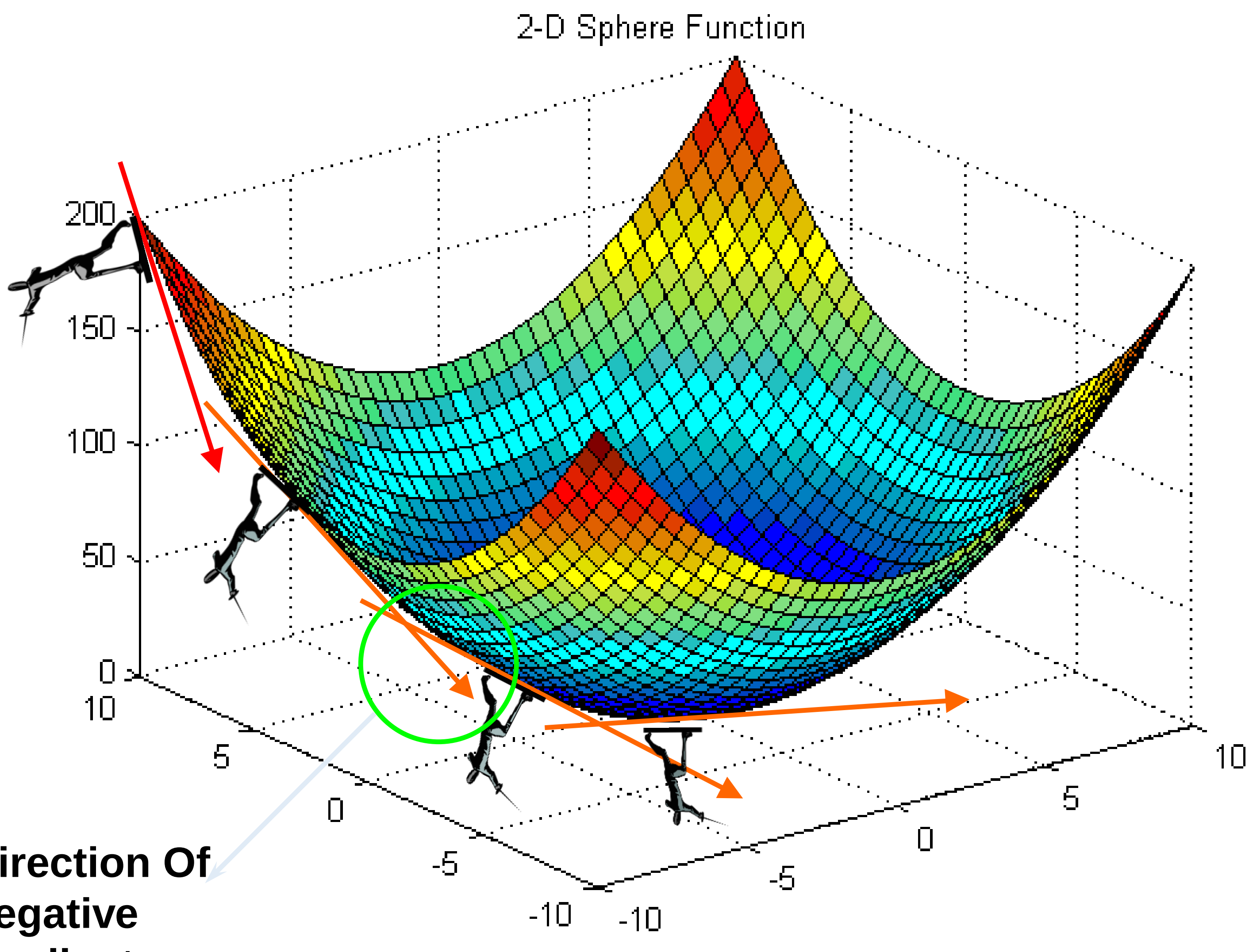
Multimodal problems

- There are a lot of interesting problems with multiple optima.



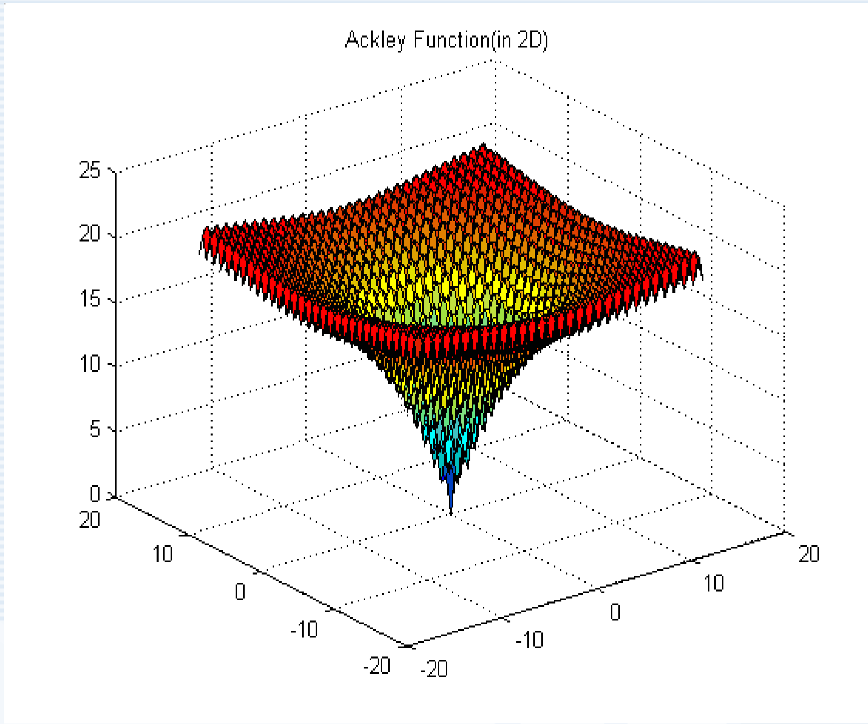
- In some problems we want to obtain a set of multiple solutions.

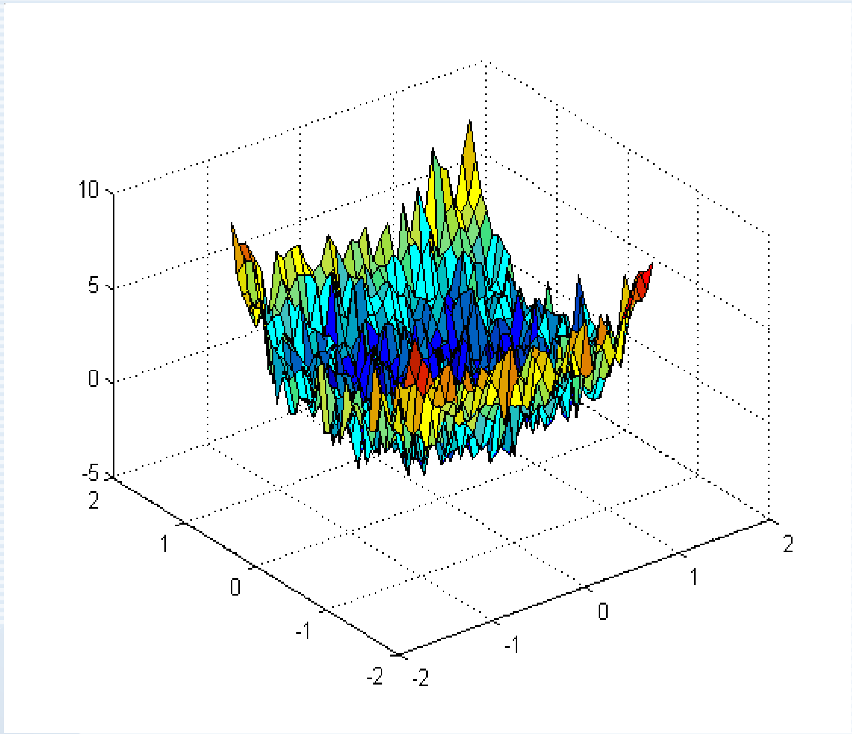




Direction Of negative gradient

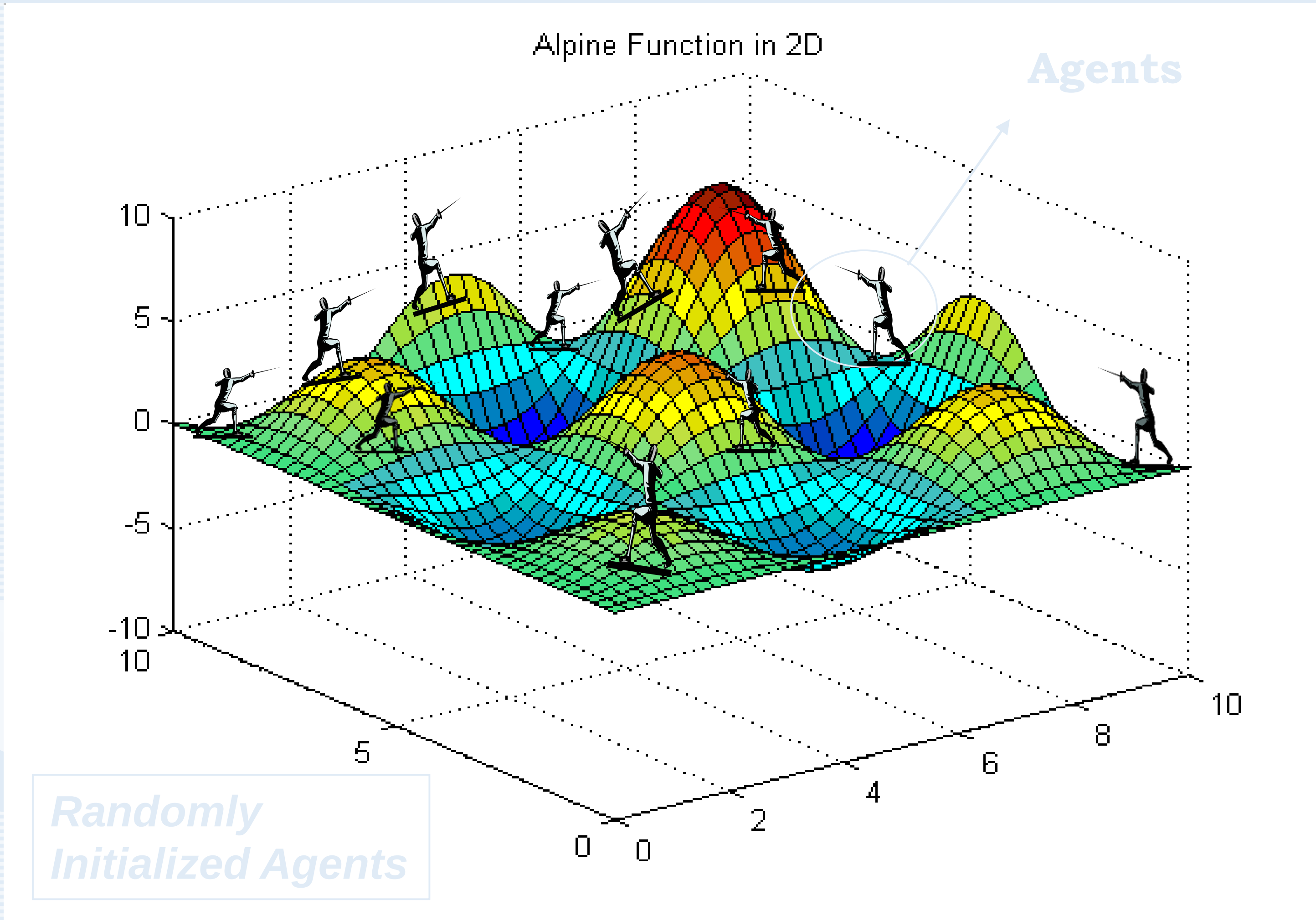
But What about these multi-modal, noisy and even discontinuous functions?



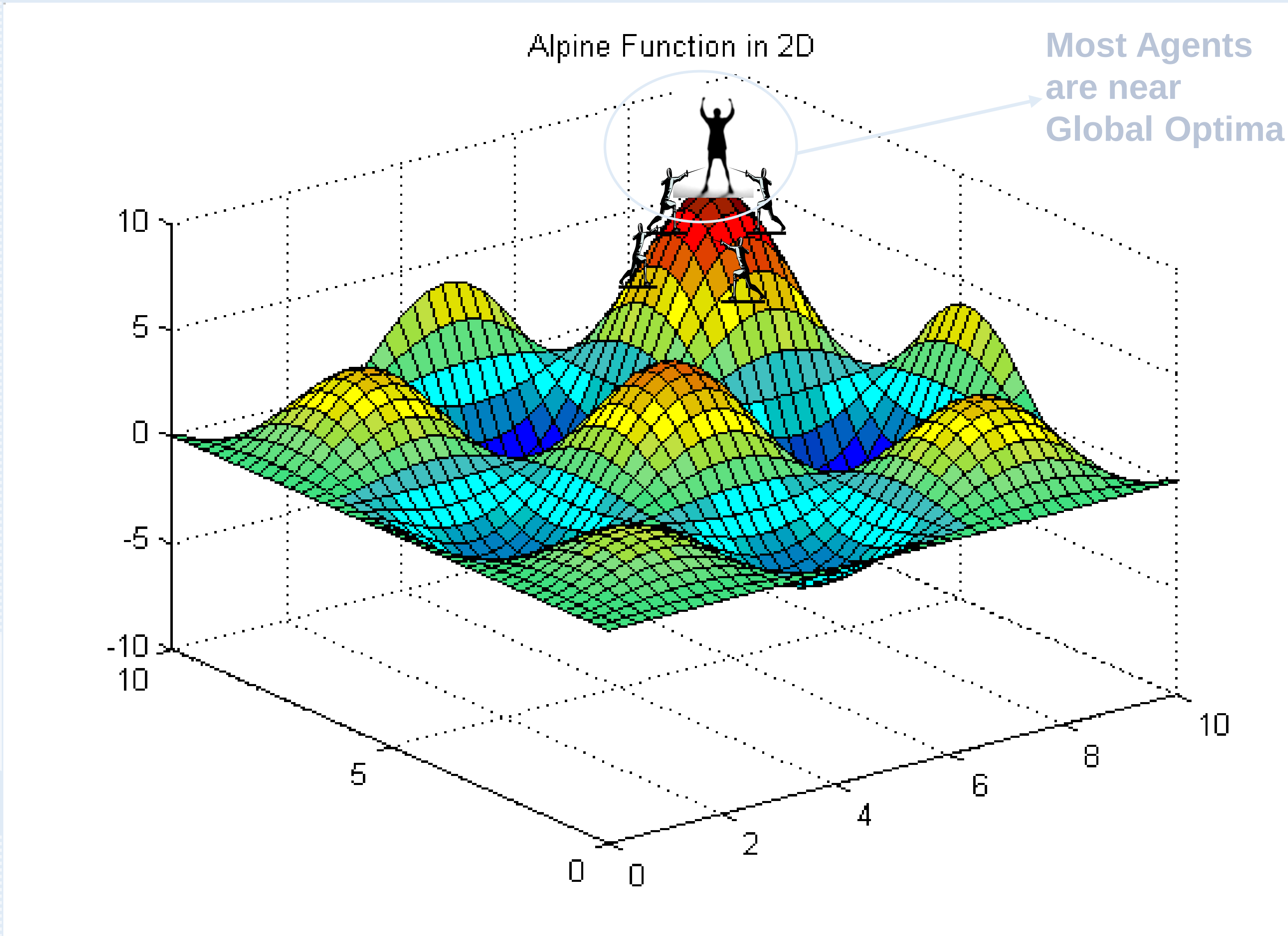


Gradient based methods get trapped in a local minima or the Function itself may be non differentiable.

Way Out: Multi-Agent Optimization in Continuous Space



After Convergence

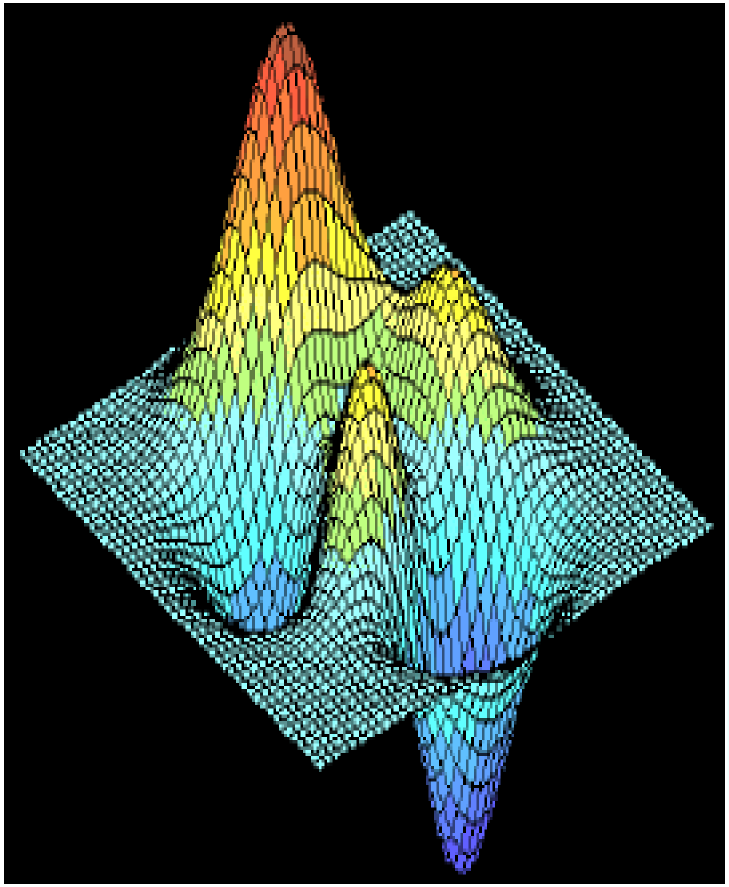


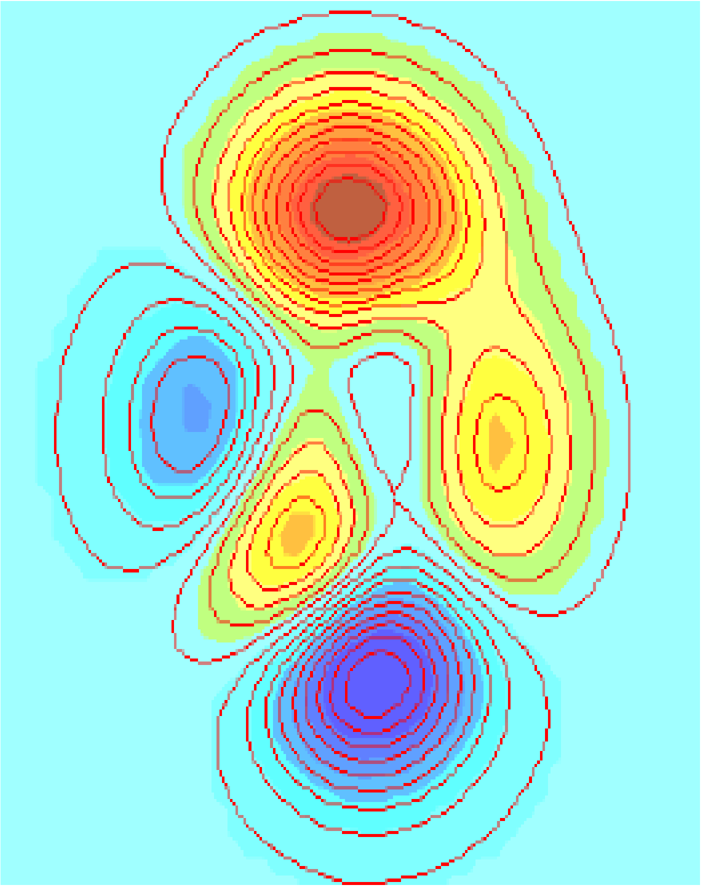
Most recent multi-agent stochastic parallel search techniques is:

Particle Swarm Optimization (PSO; Eberhart & Kennedy, IEEE Evocomp, 1995)

- **Example: Max z= f(x,y)**

$$z = f(x, y) = 3*(1-x)^2*\exp(-(x^2) - (y+1)^2) - 10*(x/5 - x^3 - y^5)*\exp(-x^2-y^2) - 1/3*\exp(-(x+1)^2 - y^2).$$

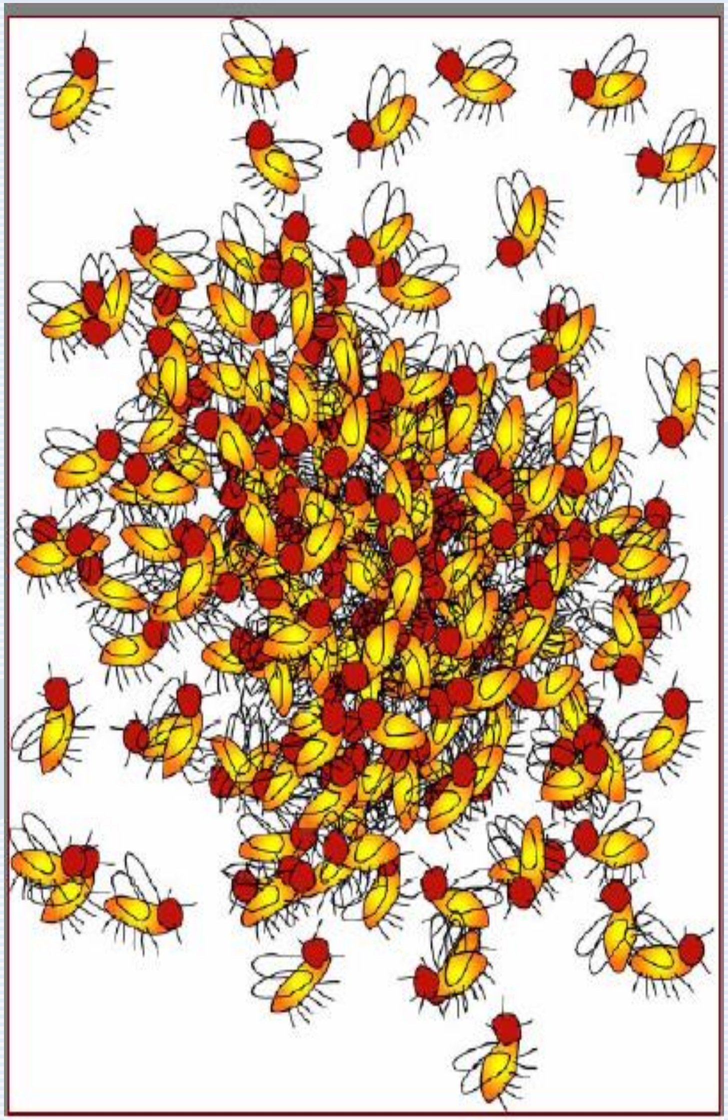




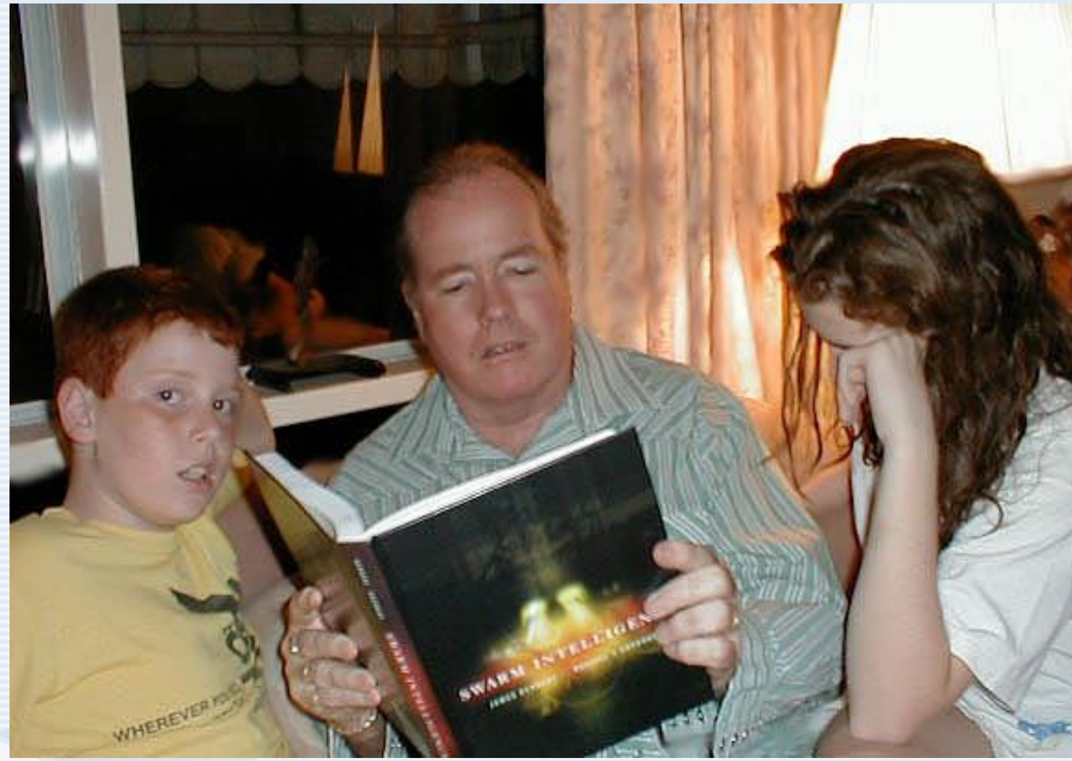
Particle Swarm Optimization (PSO)

- PSO is a robust stochastic optimization technique based on the movement and intelligence of swarms.
- PSO applies the concept of social interaction to problem solving.

SWARMS



Inventors (1995)



Russell Eberhart

(electrical engineer)

James Kennedy

(social-psychologist)

Why do animals swarm?

- Defense against predators
- Enhanced predator detection
- Minimizing chance of capture
- Enhanced foraging success
- Better chances to find a mate
- Decrease of energy consumption

- It uses a number of agents (particles) that constitute a swarm moving around in the search space looking for the best solution.
- Each particle is treated as a point in a N-dimensional space which adjusts its “flying” according to its own flying experience as well as the flying experience of other particles.

- Each particle keeps track of its coordinates in the solution space which are associated with the best solution (fitness) that has achieved so far by that particle. This value is called personal best , ***pbest***.

PSO Continues

- Another best value that is tracked by the PSO is the best value obtained so far by any particle in the neighborhood of that particle. This value is called ***gbest***.
- The basic concept of PSO lies in accelerating each particle toward its ***pbest*** and the ***gbest*** locations, with a random weighted accelaration at each time step as shown in Fig.1

PSO Contin….

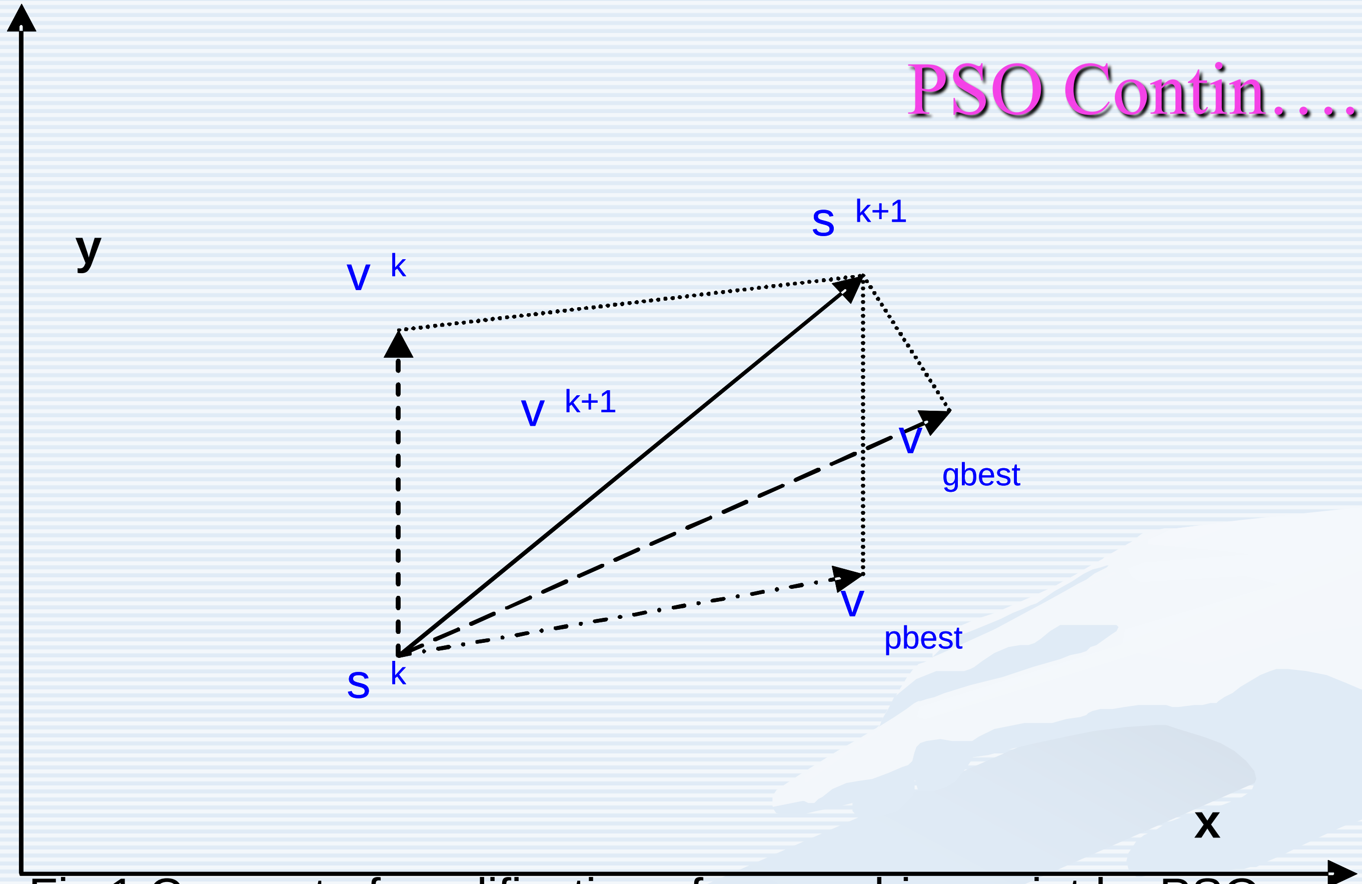


Fig.1 Concept of modification of a searching point by PSO

s^k : current searching point.

s^{k+1} : modified searching point.

v^k : current velocity.

v^{k+1} : modified velocity.

v_{pbest} : velocity based on pbest.

v_{gbest} : velocity based on gbest

Each particle tries to modify its position using the following information:

- the current positions,
- the current velocities,
- the distance between the current position and pbest,
- the distance between the current position and the gbest.

Mathematical Equation

•The modification of the particle's position can be mathematically modeled according the following equation :

$$V_i^{k+1} = wV_i^k + c_1\, rand_1(\ldots) \times (pbest_i - s_i^k) + c_2\, rand_2(\ldots) \times (gbest - s_i^k) \quad \ldots\ldots \quad (1)$$

where,

v_i^k : velocity of agent i at iteration k,

w: weighting function,

C_1 and C_2 : weighting factor,

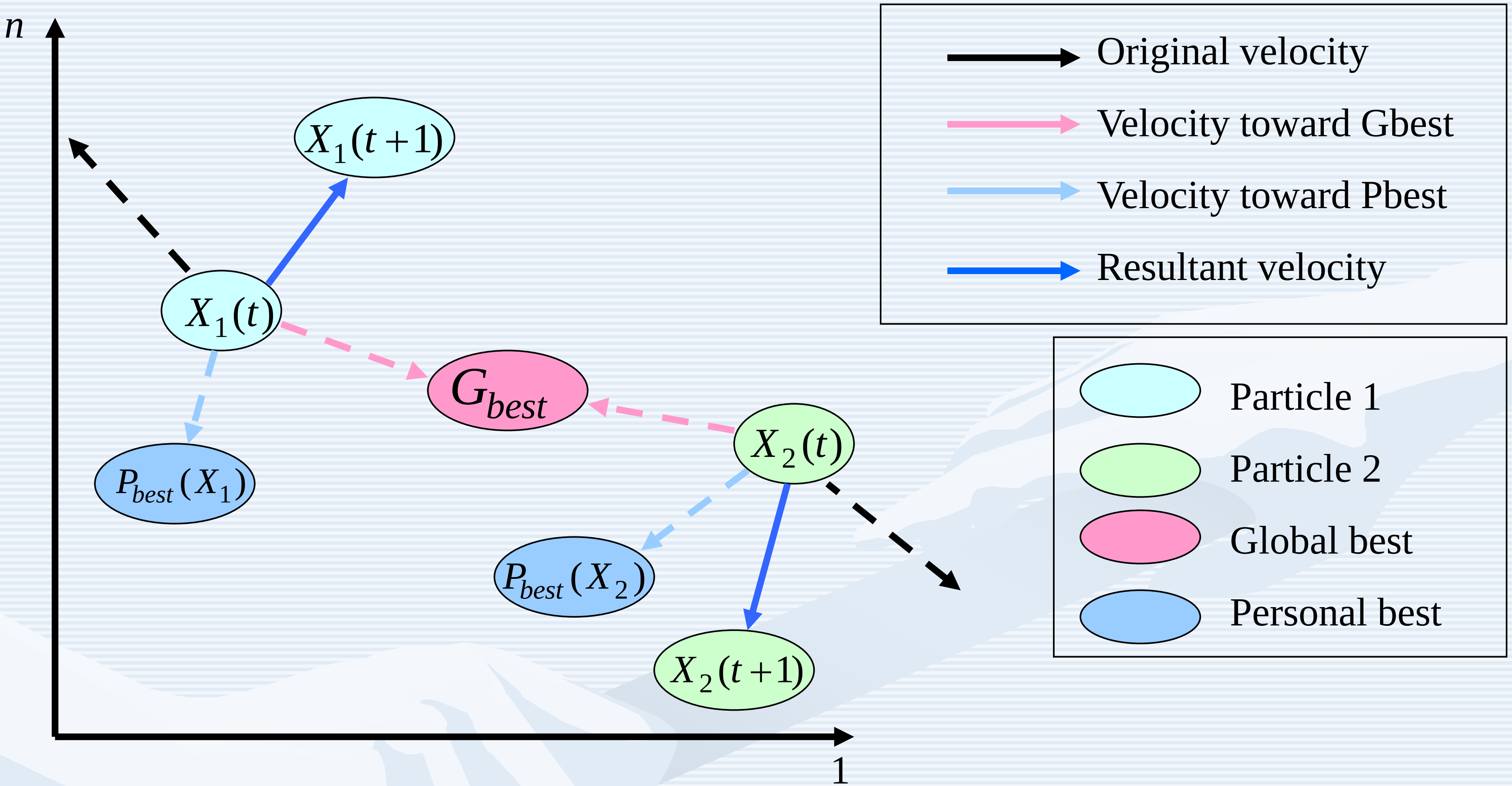
rand : uniformly distributed random number between 0 and 1,

s_i^k : current position of agent i at iteration k,

$pbest_i$: pbest of agent i,

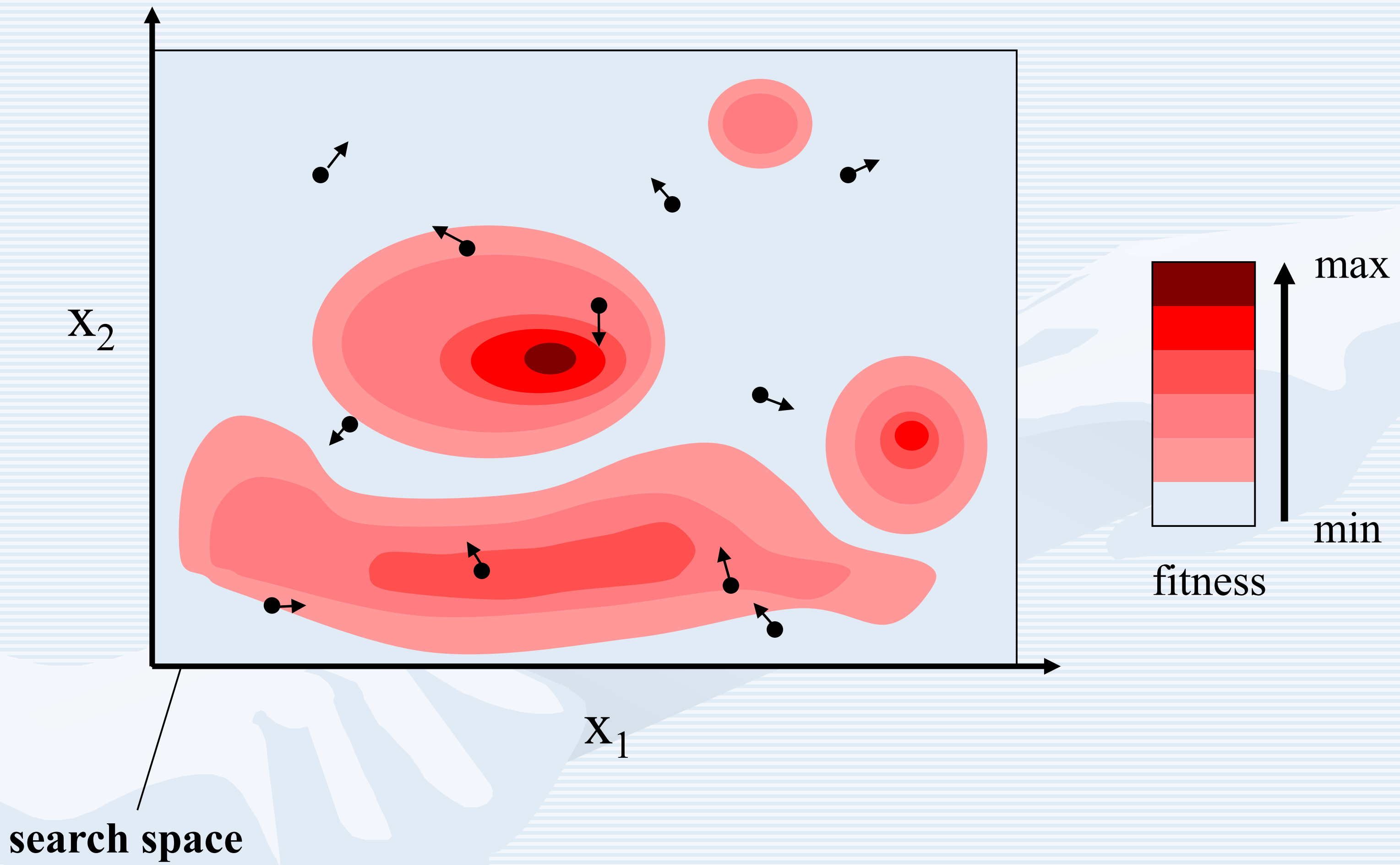
gbest: gbest of the group.

Aspects of Basic PSO (movement of particles

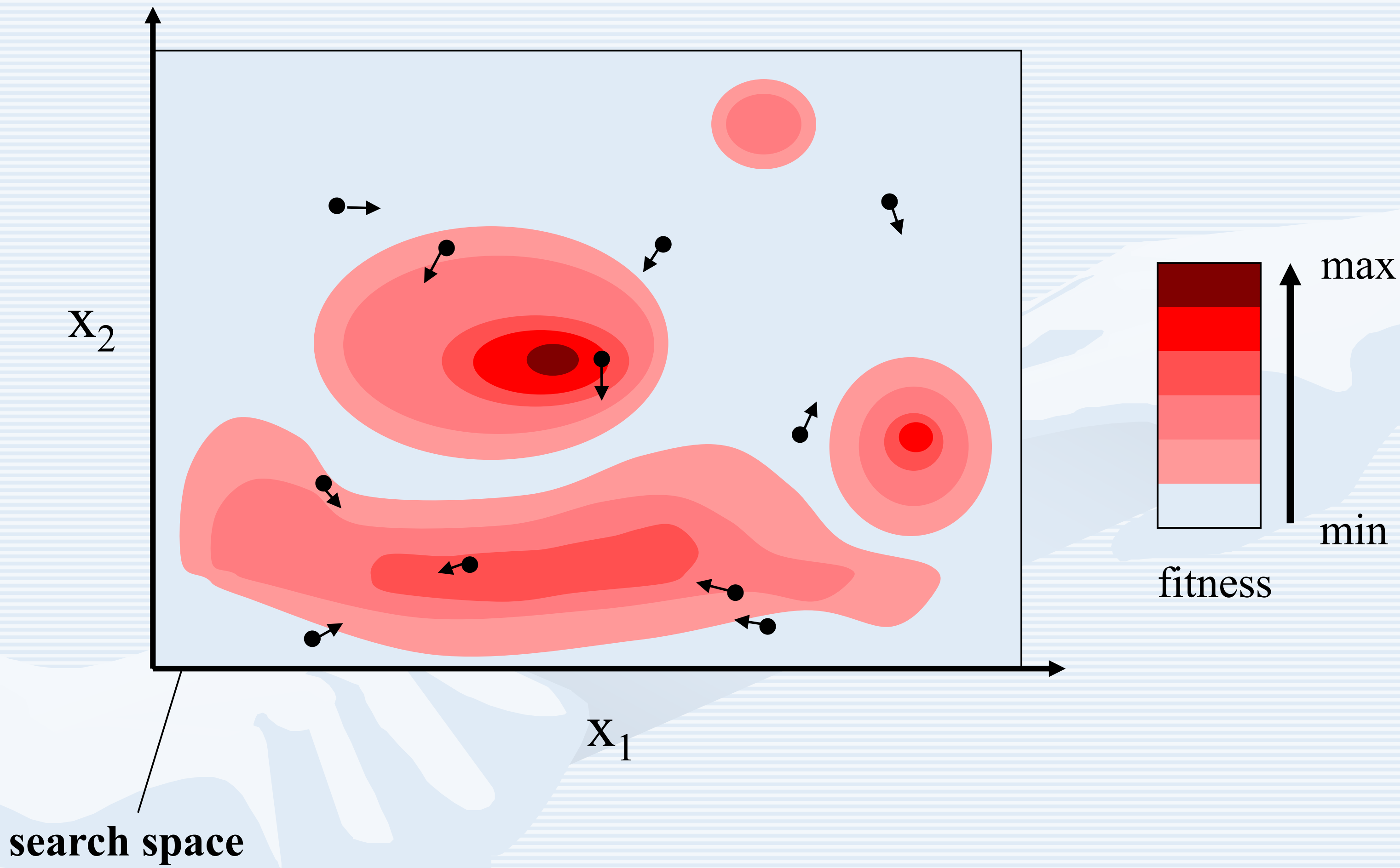


Individual particles (1 and 2) are accelerated toward the location of the global best solution (Gbest) and the location of their own personal best (Pbest) in the *n*-dimensional space.

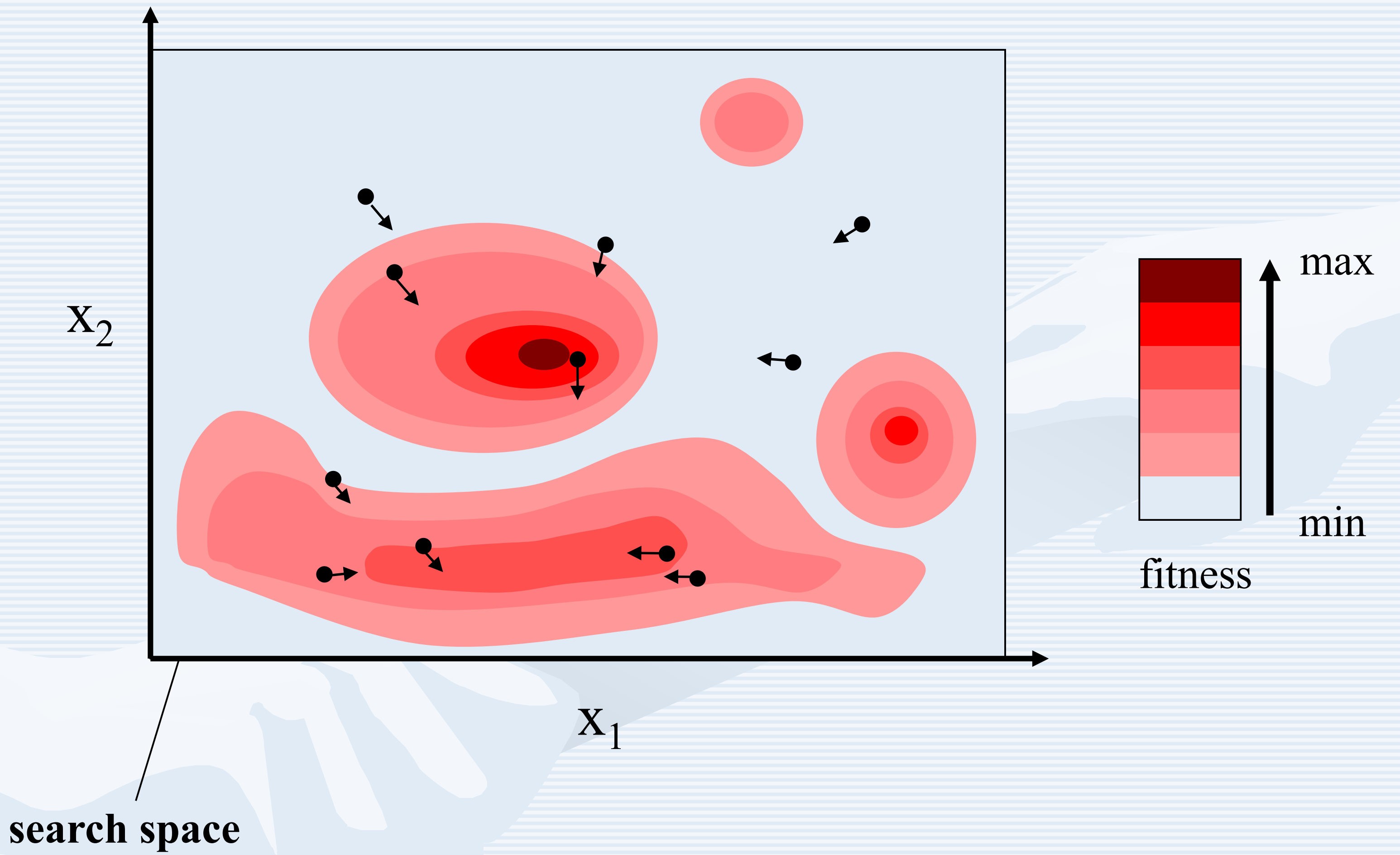
Particle Swarm Optimization – Animation



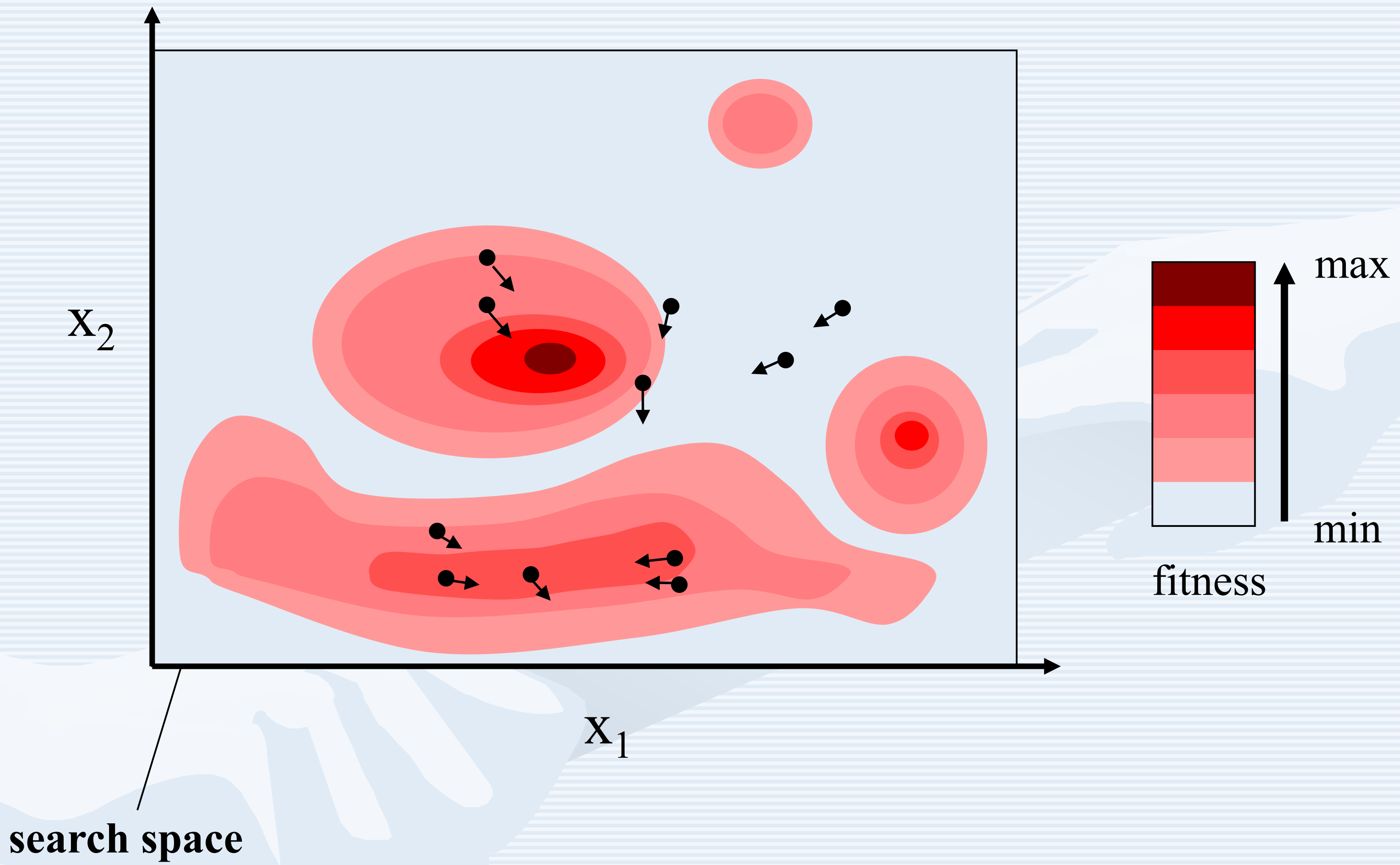
Particle Swarm Optimization – Animation



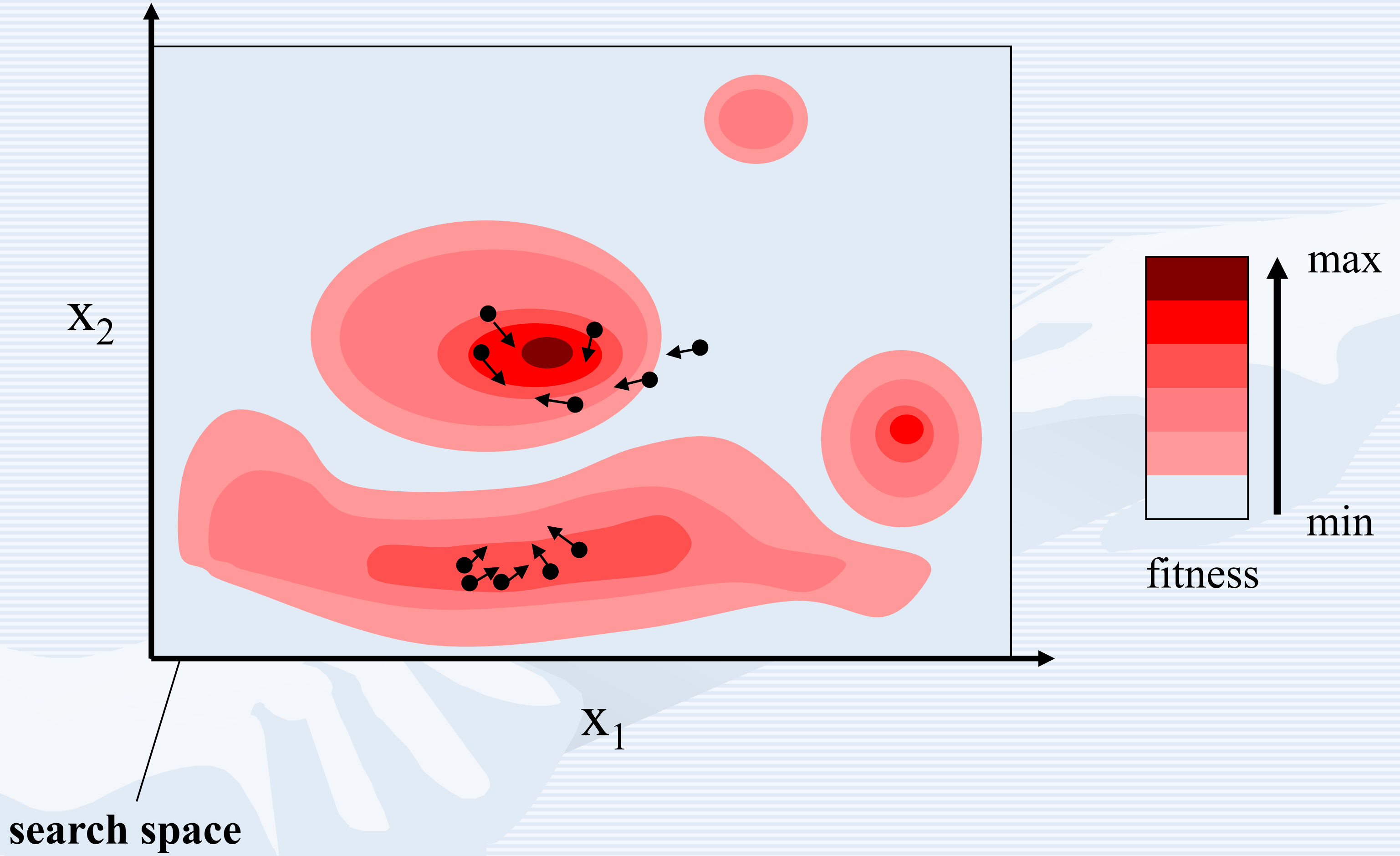
Particle Swarm Optimization – Animation



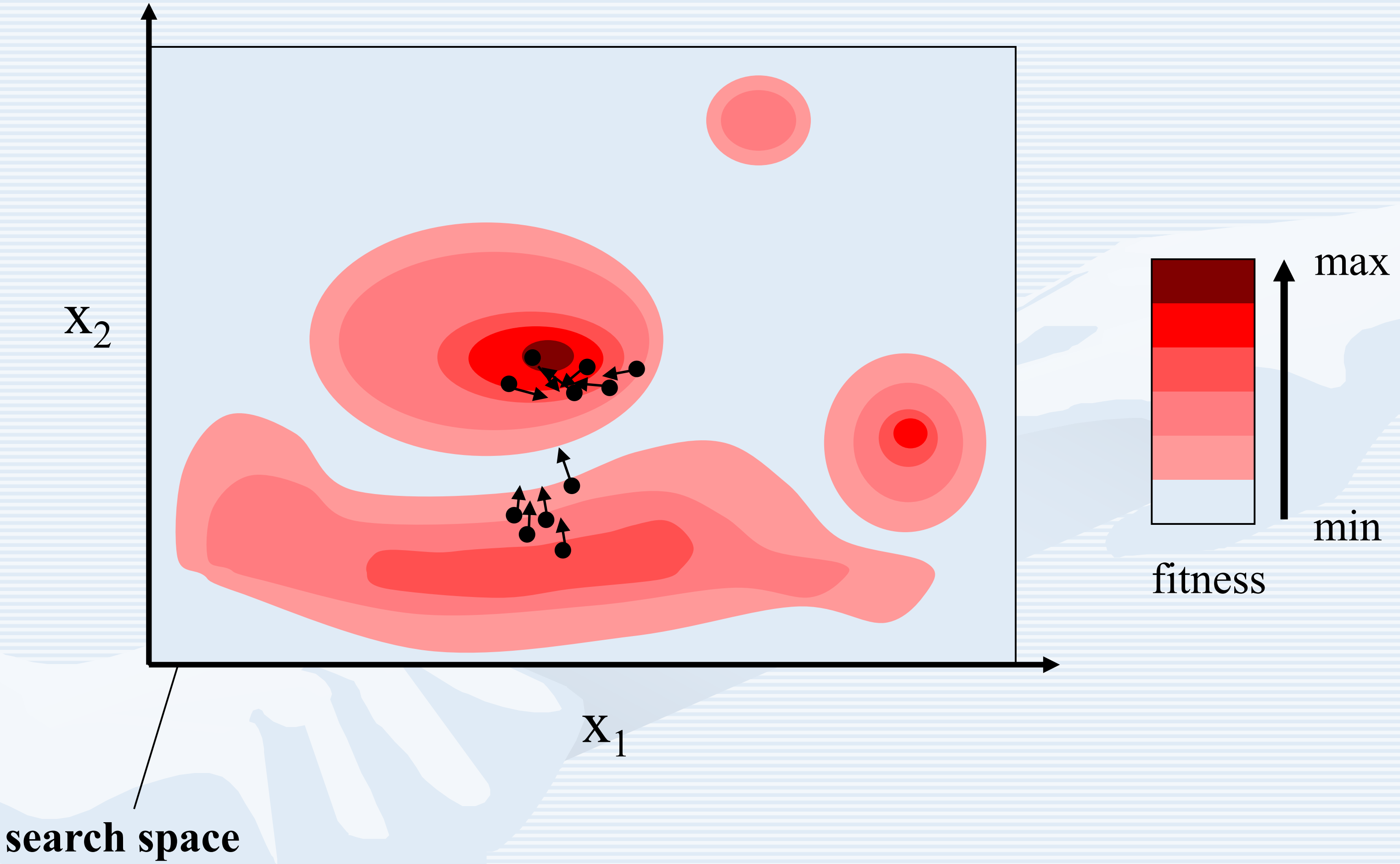
Particle Swarm Optimization – Animation



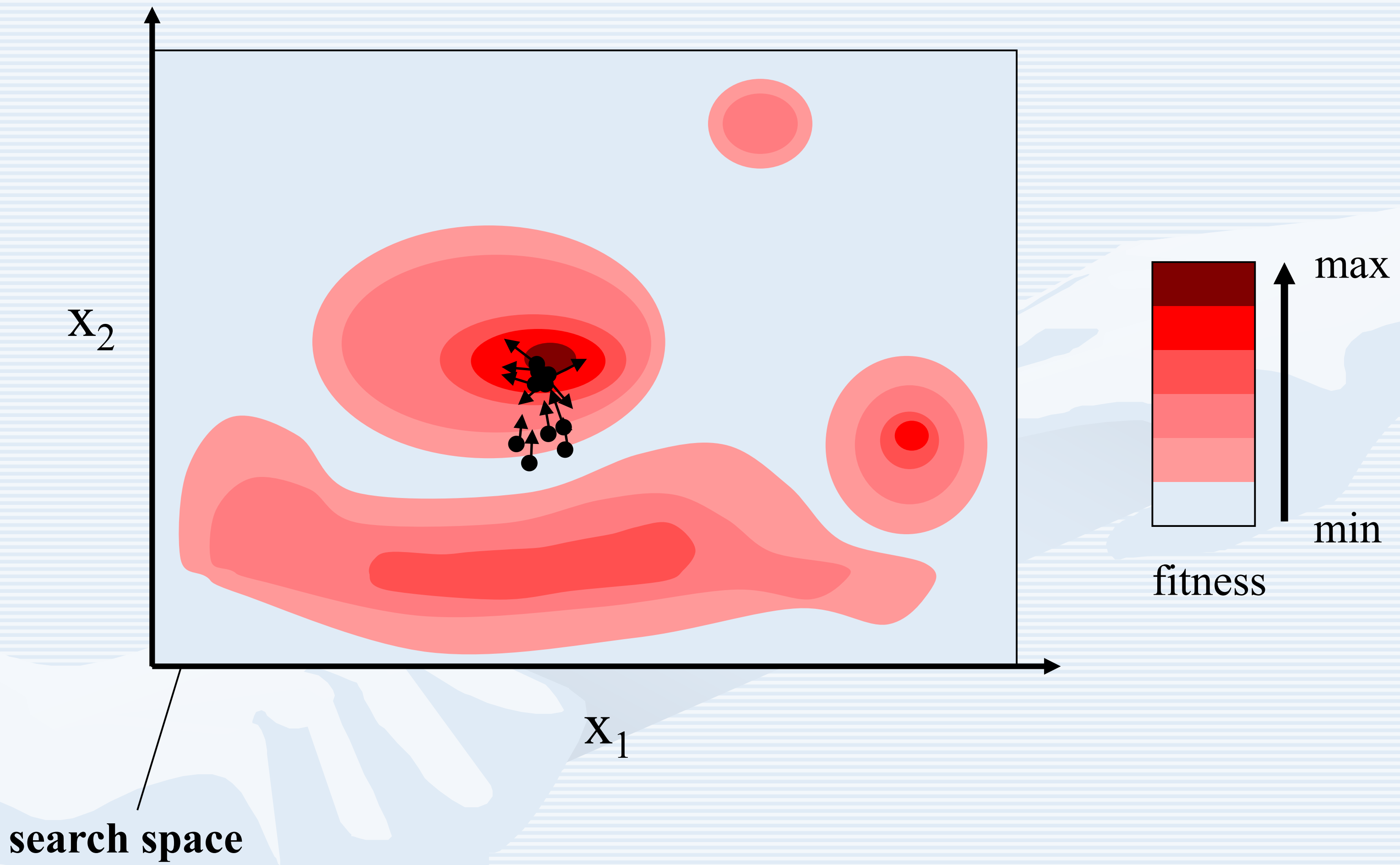
Particle Swarm Optimization – Animation



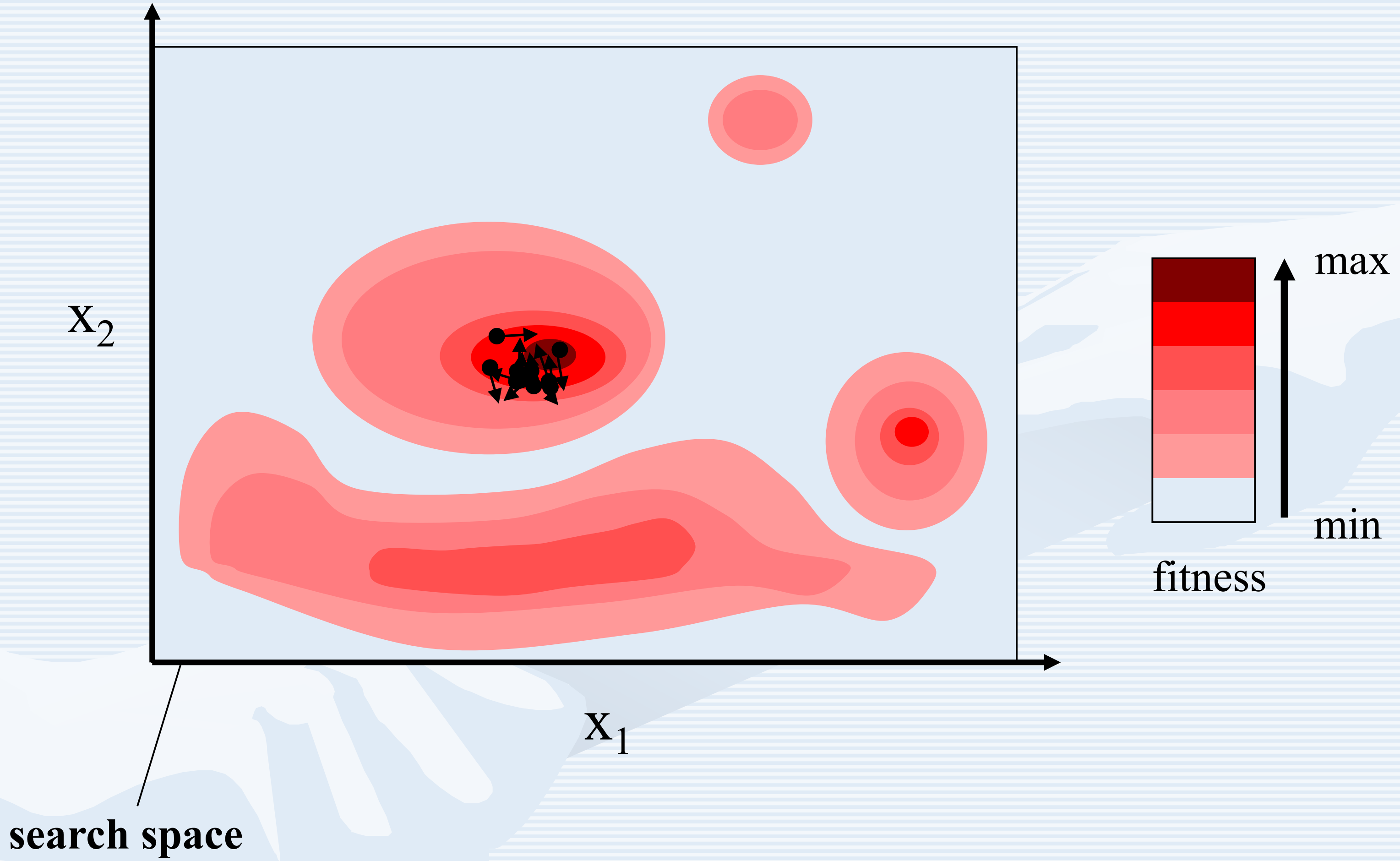
Particle Swarm Optimization – Animation



Particle Swarm Optimization – Animation



Particle Swarm Optimization – Animation



$$f(x) = x(x-8)$$

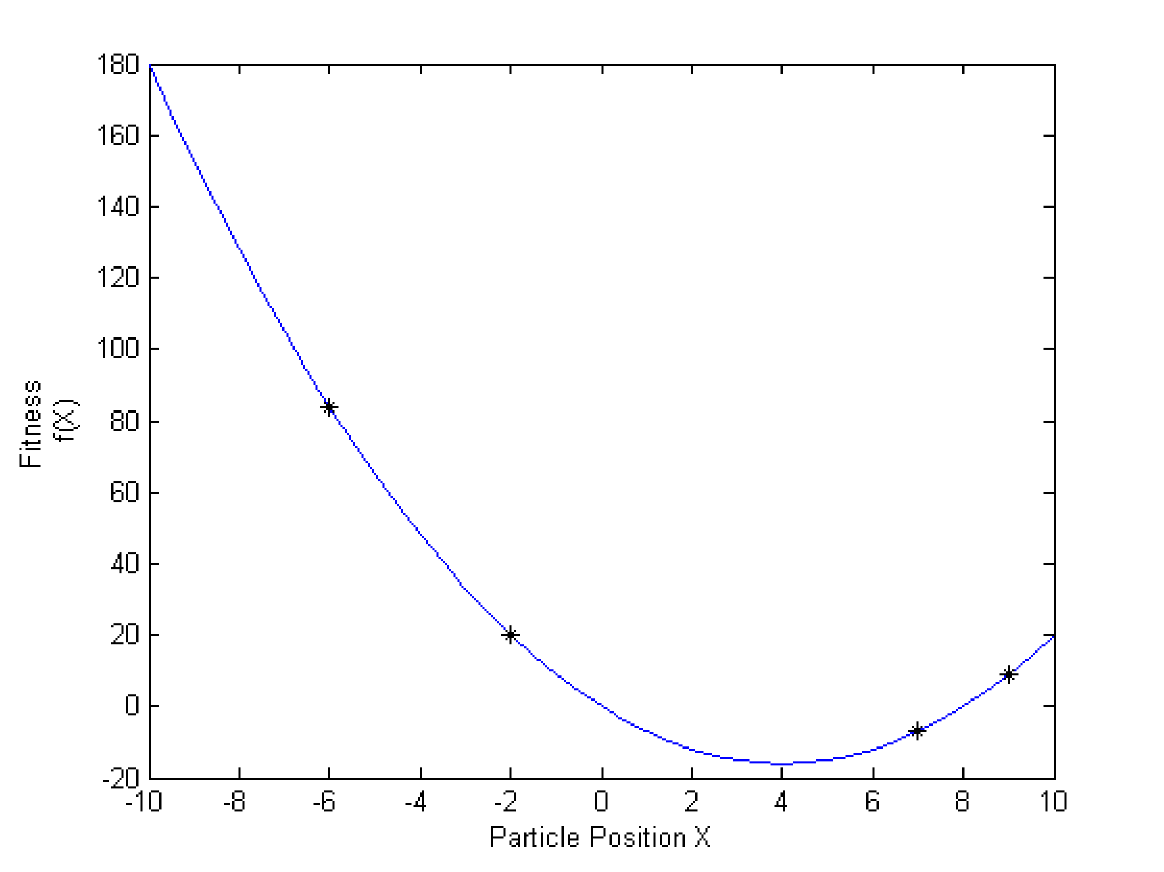
Consider a small swarm of particles for the above single dimensional function:

Initial Position and velocities of the particles at time t=0: Randomly initialized In the range (-10, 10)

Particle number	Position x(0) at t = 0	Velocity v at t = 0	f (x)
1	7	3	-7
2	-2	5	20
3	9	6	9
4	-6	-4	84

So, the fittest particle is particle 1 and we set $P_{gb} = 7$
And $P_{lb} = X_i$

Initial Distribution of the Particles over the fitness landscape



Change in position of the particles in next iteration:

$$V_i(t+1)=\varphi.V_i(t)+C_1.rand(0,1).(P_{lb}-X_i(t))+C_2.rand(0,1).(P_{gb}-X_i(t))$$

$$X_i(t+1)=X_i(t)+V_i(t+1)$$

For this small scale PSO problem, we set $C_1 = C_2 = 2.0, \omega = 0.5$

Particle 1>

$V_1(1) = 0.5*3 + 2*0.6*(7 – 7) + 2*0.4*(7 – 7) = 1.5$

$X_1(1) = 7+1.5 = 8.5$

Fitness f (X(1)) =4.25

Particle 2>

$V_2(1) = 0.5*5 + 2*0.3*(-2 + 2) + 2*0.4*(7 – (-2)) = 6.5$

$X_2(1) = -2+6.5 = 3.5$

Fitness f (X(1)) =-9.75

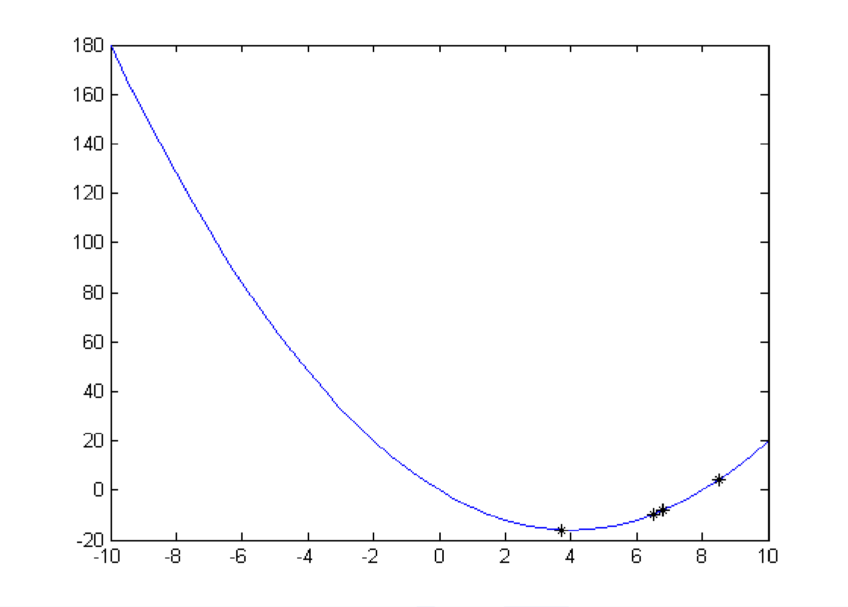
Particle 3>
$V_3(1) = 0.5*6 + 2*0.8*(9 - 9) + 2*0.95*(7 – 9) = -0.8$
$X_3(1) = 6 – (-0.8) = 6.8$
Fitness f (X(1)) =-8.16

Particle 4>
$V_4(1) = 0.5*(-4) + 2*0.38*(-6 + 6) + 2*0.45*(7 – (-6)) = 9.7$
$X_4(1) = -6 + 9.3 = 3.7$
Fitness f (X(1)) =-15.91

Here we go for the next iteration:

Particle number	Position at t = 1	Velocity at t = 1	f (x)	P_{lb} for t = 2	P_{gb} for t = 2
1	8.5 (at t = 0, 7)	1.5 (at t = 0, 3)	4.25 (at t = 0, -7)	7	3.7
2	6.5 (-2)	3.5 (5)	-9.75 (20)	6.5	3.7
3	6.8 (9)	-0.8 (6)	-8.16 (9)	6.8	3.7
4	3.7 (-6)	9.7 (-4)	-15.91(84)	3.7	3.7

Distribution of the Particles over the fitness landscape at t = 1

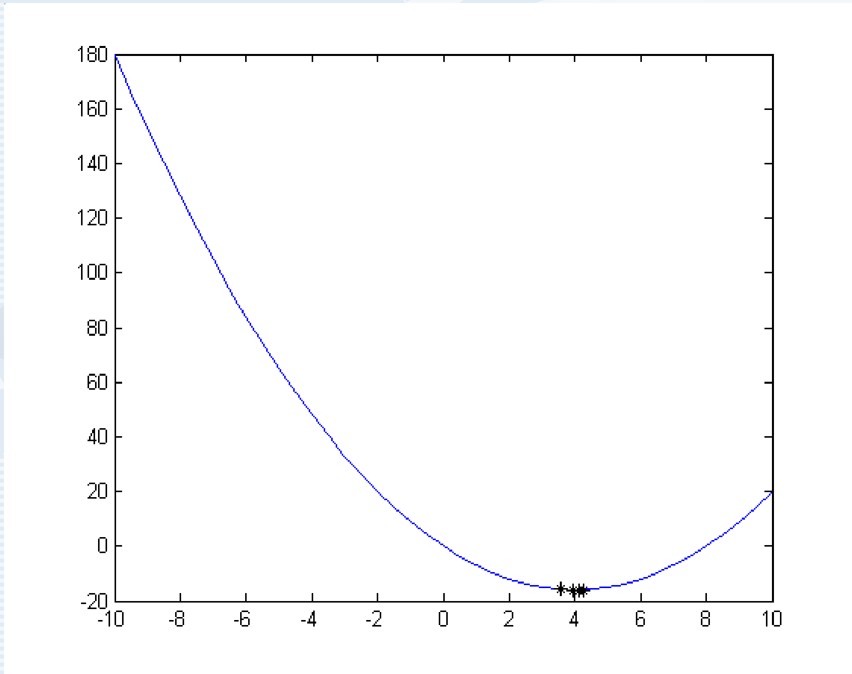


Distribution of the Particles over the fitness landscape at t = 5

Best particle at t = 5

P_{gb} = 3.95

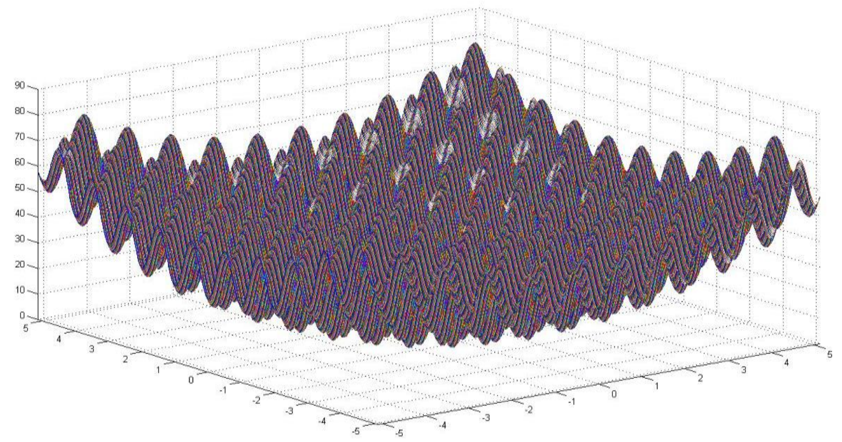
$f(P_{gb})$ = -15.99



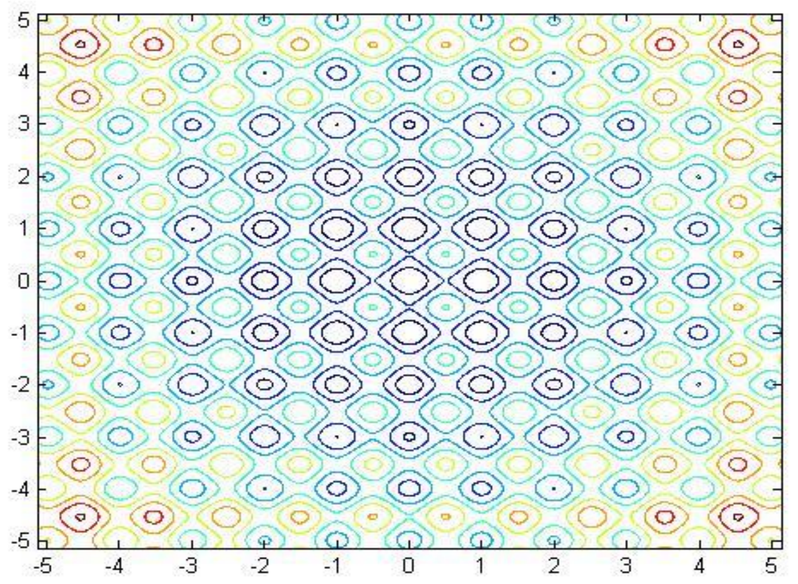
Rastrigin Function (n=2)

$$f(x) = An + \sum_{i=1}^{n} [x_i^2 + A\cos(2\pi x_i)]$$

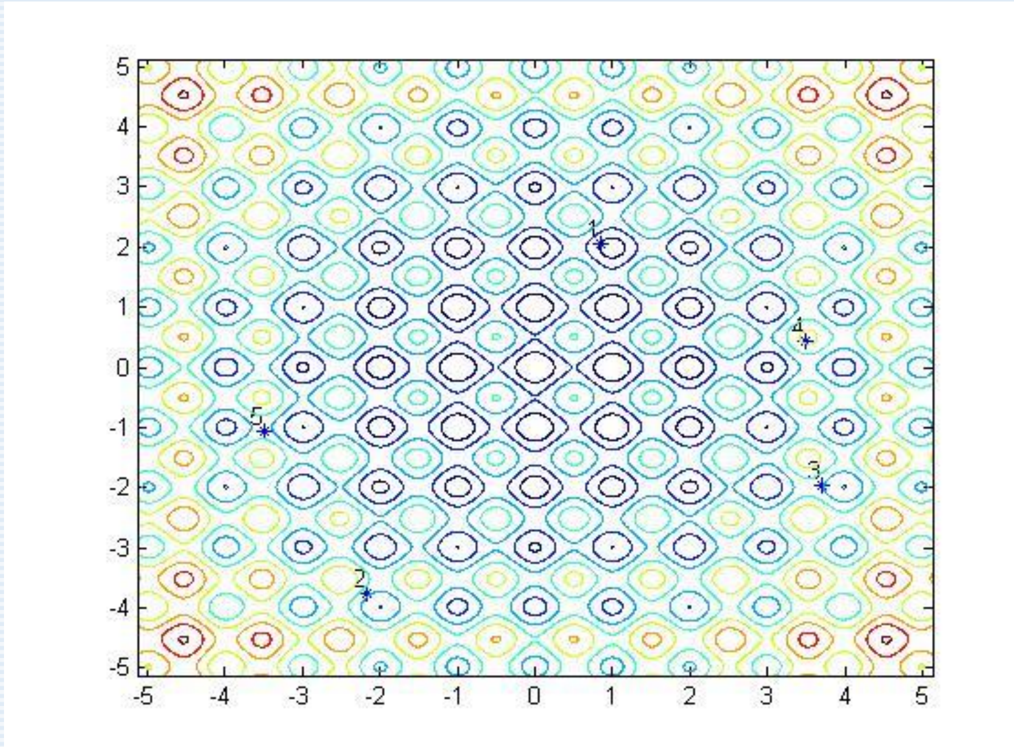
where, $A = 10, x_i \in [-5.12, 5.12]$



Parameters values	
Parameters	Value
φ_1, φ_2	2
ω	0.5
$v_{i,d}(1)$	0

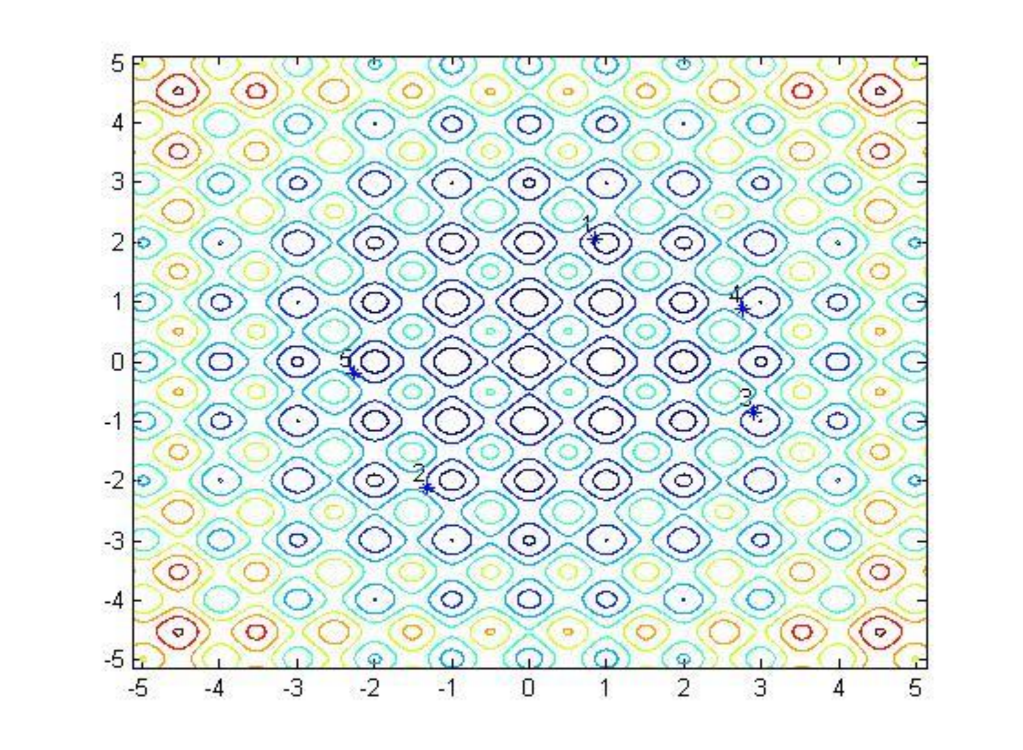


Iteration 1



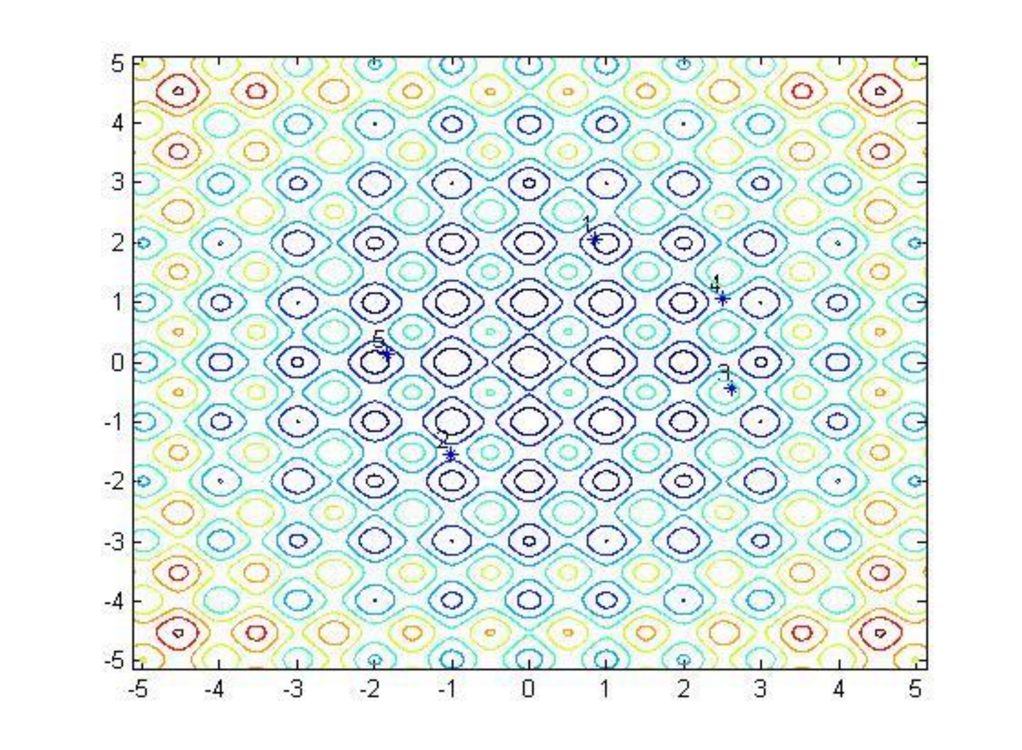
Iteration no = 1 (rand1 = 0.07, rand2 = 0.14)											
Particle i	Initial particle		f(x)	Individual best		Global best		Velocity		Updated particle	
	x1	x2		x1	x2	x1	x2	v1	v2	x1	x2
1	0.85	2.06	9.79	0.85	2.06			0	0	0.85	2.06
2	-2.17	-3.75	33.95	-2.17	-3.75			0.85	1.63	-1.32	-2.12
3	3.7	-1.97	30.84	3.7	-1.97	0.85	2.06	-0.8	1.13	2.9	-0.84
4	3.49	0.44	51.65	3.49	0.44			-0.74	0.45	2.75	0.89
5	-3.49	-1.07	34.26	-3.49	-1.07			1.22	0.88	-2.27	-0.19

Iteration 2



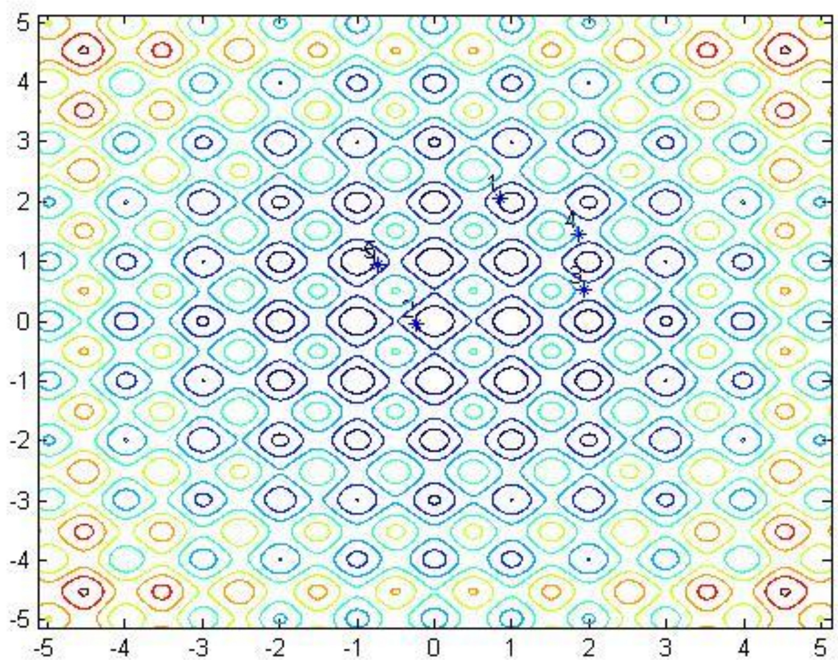
Iteration no = 2 (rand1 = 0.63, rand2 = 0.05)											
Particle i	Initial particle		f(x)	Individual best		Global best		Velocity		Updated particle	
	x1	x2		x1	x2	x1	x2	v1	v2	x1	x2
1	0.85	2.06	9.79	0.85	2.06	0.85	2.06	0	0	0.85	2.06
2	-1.32	-2.12	23.2	-1.32	-2.12			0.3	0.58	-1.02	-1.54
3	2.9	-0.84	15.67	2.9	-0.84			-0.29	0.4	2.61	-0.44
4	2.75	0.89	23.92	2.75	0.89			-0.26	0.16	2.49	1.05
5	-2.27	-0.19	22.76	-2.27	-0.19			0.43	0.31	-1.84	0.12

Iteration 3



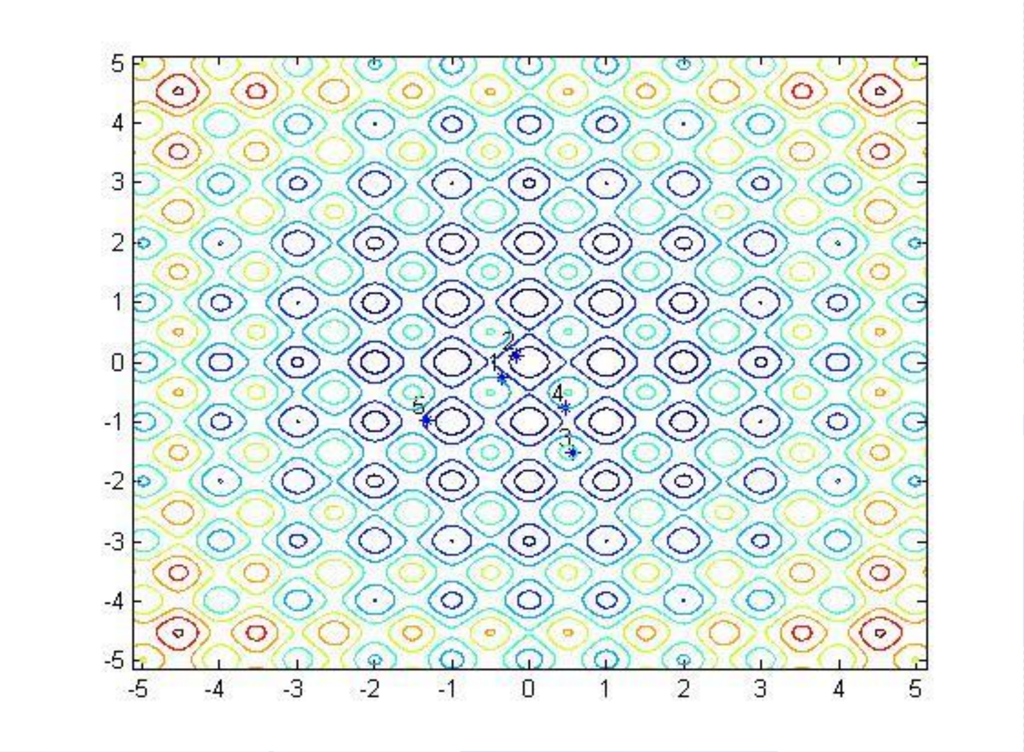
Iteration no = 3 (rand1 = 0.1, rand2 = 0.2)											
Particle i	Initial particle		f(x)	Individual best		Global best		Velocity		Updated particle	
	x1	x2		x1	x2	x1	x2	v1	v2	x1	x2
1	0.85	2.06	9.79	0.85	2.06	0.85	2.06	0	0	0.85	2.06
2	-1.02	-1.54	23.18	-1.02	-1.54			0.78	1.5	-0.24	-0.04
3	2.61	-0.44	44.01	2.9	-0.84			-0.68	0.96	1.93	0.52
4	2.49	1.05	27.77	2.75	0.89			-0.63	0.39	1.86	1.44
5	-1.84	0.12	10.75	-1.84	0.12			1.12	0.81	-0.72	0.93

Iteration 4



Iteration no = 4 (rand1 = 0.56, rand2 = 0.55)												
Particle i	Initial particle		f(x)	Individual best		Global best		Velocity		Updated particle		
	x1	x2		x1	x2	x1	x2	v1	v2	x1	x2
1	0.85	2.06	9.79	0.85	2.06			-1.2	-2.31	-0.35	-0.25
2	-0.24	-0.04	9.75	-0.24	-0.04			0.08	0.15	-0.16	0.11
3	1.93	0.52	24.87	2.9	-0.84	-0.24	-0.04	-1.37	-2.04	0.56	-1.52
4	1.86	1.44	28.46	2.75	0.89			-1.38	-2.21	0.48	-0.77
5	-0.72	0.93	14.21	-1.84	0.12			-0.61	-1.89	-1.33	-0.96

Iteration 5

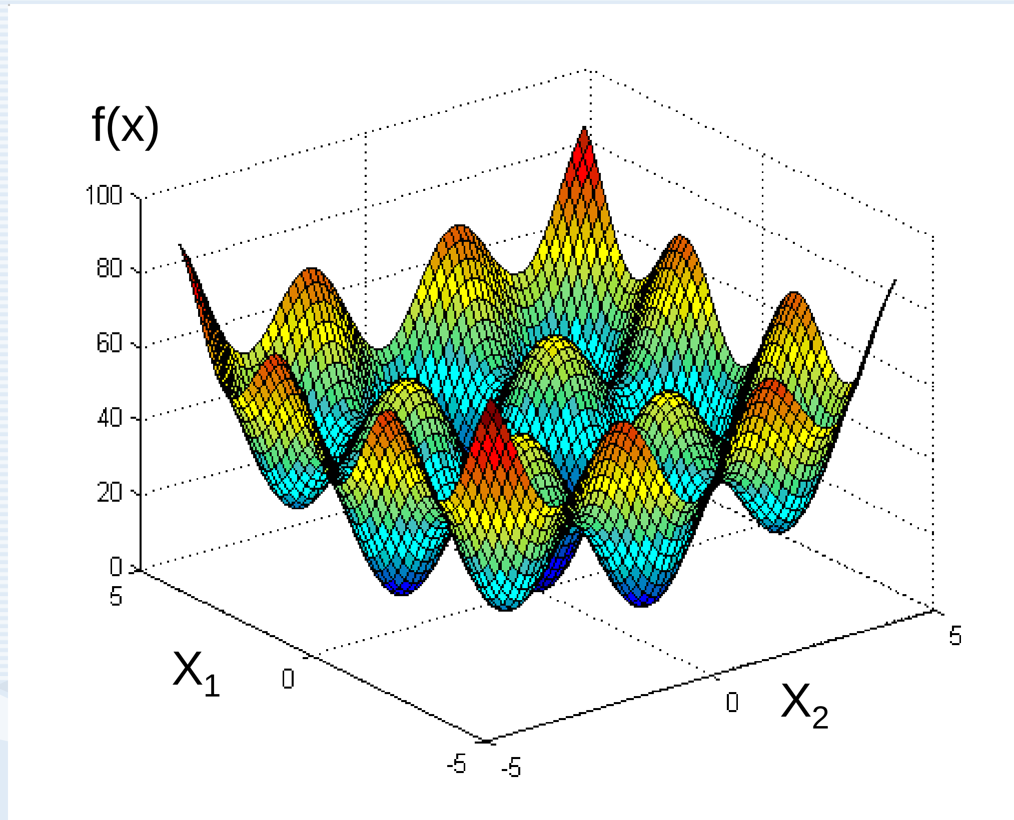


Iteration no = 5 (rand1 = 0.41, rand2 = 0.38)											
Particle i	Initial particle		f(x)	Individual best		Global best		Velocity		Updated particle	
	x1	x2		x1	x2	x1	x2	v1	v2	x1	x2
1	-0.35	-0.25	26.06	0.85	2.06	-0.16	0.11	1.01	1.94	0.66	1.69
2	-0.16	0.11	6.97	-0.16	0.11			0.01	0.02	-0.15	0.13
3	0.56	-1.52	41.84	2.9	-0.84			1.23	1.59	1.79	0.07
4	0.48	-0.77	29.49	2.75	0.89			1.24	1.81	1.72	1.04
5	-1.33	-0.96	17.82	-1.84	0.12			0.41	1.51	-0.92	0.55

Optimization by PSO: Egg crate Function

Minimize $$f(\vec{x}) = x_1^2 + x_2^2 + 25(\sin^2 x_1 + \sin^2 x_2)$$

- *Known global minima at [0,0] and optimum function value 0*



- PSO parameters:
  - Swarm Size = 30
  - Inertia, w = 0.5 (static)
  - Self Confidence, c_1 = 1.5
  - Swarm Confidence, c_2 = 1.5
  - Stopping Tolerance, ε = 0.001

Eggcrate Function Optimization by PSO

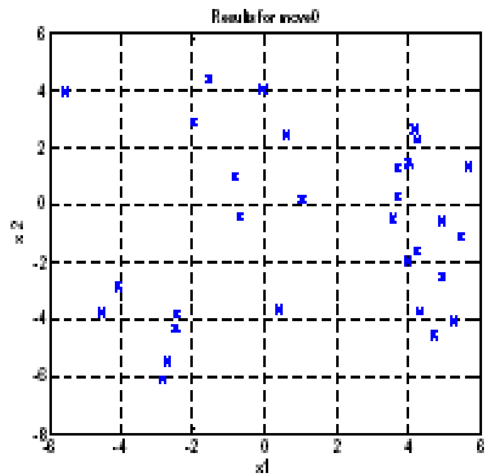
Position of the particles on a 2D parameter Space at different instances

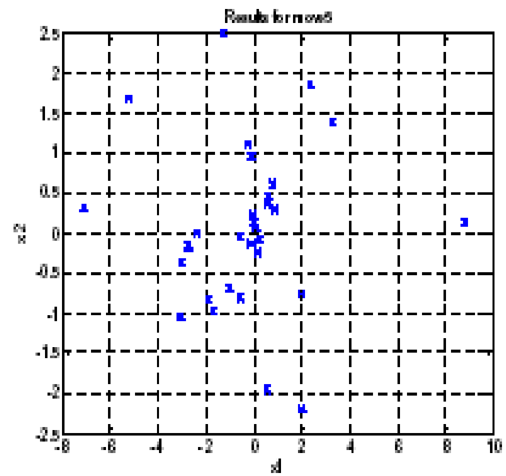
Statistics

Min = 15.1

Mean = 49.9

Max = 92.6





Statistics

Min = 0.25

Mean = 26.5

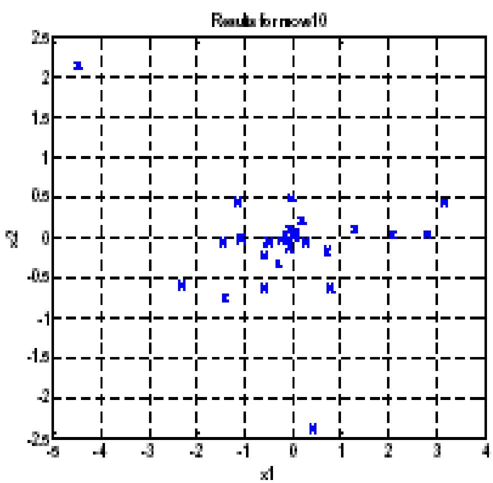
Max = 86.6

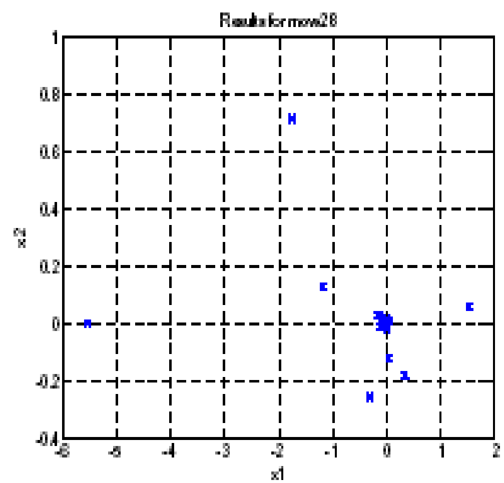
Statistics

Min = 0.02

Mean = 13.6

Max = 66.5





Statistics

Min = 0.3E-5

Mean = 4.69

Max = 41.9

Inertia weight [32]

- to control exploration and exploitation
- controls the momentum
- velocity update changes to

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] \\ + c_2 r_{2j}(t)[\hat{y}_j(t) - x_{ij}(t)]$$

- for $w \geq 1$
  - velocities increase over time
  - swarm diverges
  - particles fail to change direction towards more promising regions
- for $0 < w < 1$
  - particles decelerate
  - convergence also dependent on values of c_1 and c_2
- exploration–exploitation
  - large values – favor exploration
  - small values – promote exploitation
- problem-dependent

Constriction Factor

Clerc Introduced Constriction Factor x

$$V_i^{k+1} = x\,[V_i^k + \phi_1 \text{ x } (pbest_i - s_i^k) + \phi_2 \text{ x } (gbest - s_i^k)] \ldots\ldots \quad (2)$$

$$X = 2 \,/\, [2 - \phi - sqrt(\phi^2 - 4\,\phi)]$$

$$\Phi = \phi_1 + \phi_2$$

$\Phi < 4$ slow convergence > 4 faster convergence

PSO Contin….

The following weighting function is usually utilized

$$w = wMax-[(wMax-wMin) \times iter]/maxIter \qquad (2)$$

where wMax= initial weight,

wMin = final weight,

maxIter = maximum iteration number,

iter = current iteration number.

$$s_i^{k+1} = s_i^k + V_i^{k+1} \qquad (3)$$

Comments on the Inertial weight factor:

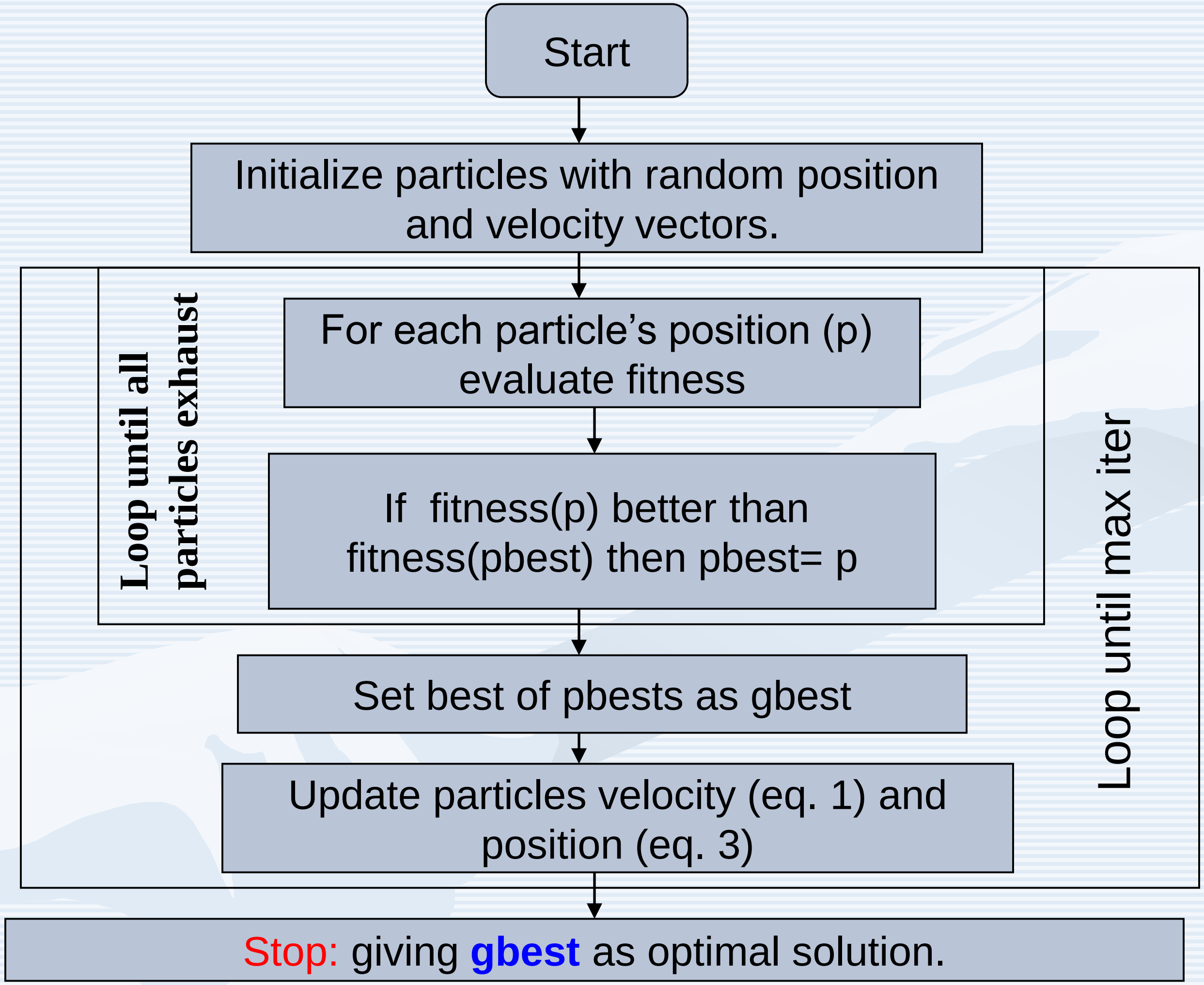
A large inertia weight (*w*) facilitates a global search while a small inertia weight facilitates a local search.

By linearly decreasing the inertia weight from a relatively large value to a small value through the course of the PSO run gives the best PSO performance compared with fixed inertia weight settings.

Larger w ----------- greater global search ability

Smaller w ------------ greater local search ability.

PSO Flow chart



Variants of PSO

- Discrete PSO ……………… can handle discrete binary variables
- MINLP PSO………… can handle both discrete binary and continuous variables.
- Hybrid PSO…………. Utilizes basic mechanism of PSO and the natural selection mechanism, which is usually utilized by EC methods such as GAs.

How to Modify the Algo

- Ratnaweera, A., Halgamuge, S.K. and Watson, H.C. (2004)
  'Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients', *IEEE Transactions on Evolutionary Computation, Vol. 8, No. 3, pp.240–255*

- **$C_1(t) = 2.5 - 2 * (t / Max\_iter)$** [Cognitive Learning Factor]
-
- **$C_2(t) = 0.5 + 2 * (t / Max\_iter)$** [Social Coefficient]

- **Yamaguchi, T. and Yasuda, K. (2006) 'Adaptive particle swarm optimization – self-coordinating mechanism with updating information',** ***Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics*****, pp.2303–2308, IEEE CS Press.**

- $C_2(t+1) = C_2(t) + \alpha * (b - C_2(t))$

α *represents the influence about the previous social* parameter,

b *is a statistical variable*

Ge, Y. and Rubo, Z. (2005) 'An emotional particle swarm optimization algorithm', *Advances in Natural Computation, Lecture Notes in Computer Science, Vol. 3612, pp.553–561,* Springer-Verlag Berlin, Germany.

- Emotion Based
- If Particle is Sad $C_2(t+1) = C_2(0) \times (r_k / r_g)$
- If Particle is Joyful $C_2(t+1) = C_2(0) \times (r_k)$

Anti-predatory PSO

- The APSO is developed by splitting both the cognitive and social behaviors of the classical PSO into two parts
- The cognitive behavior is divided into good experience (memory about previous best experience) and bad experience components (memory about previous worst experience).
- The social behavior of the classical PSO is split into global good experience and global bad experience components.

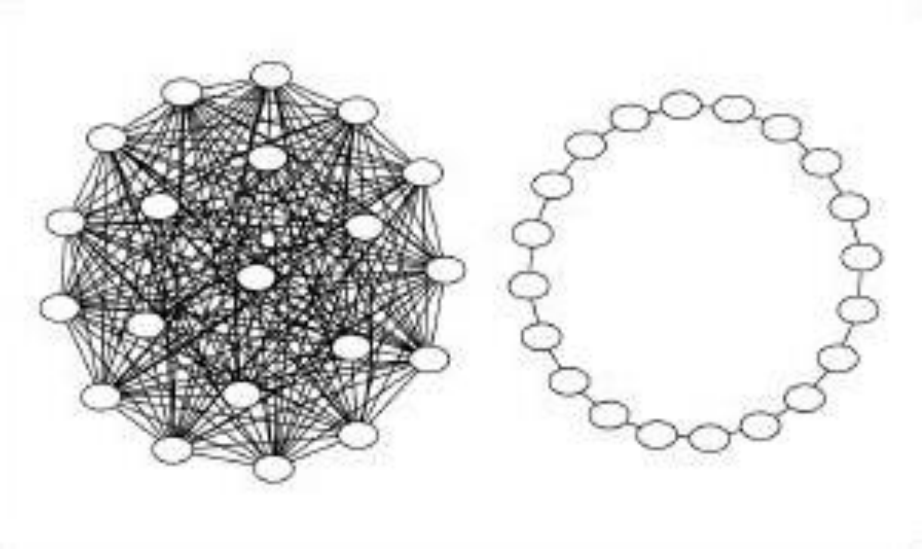
Neighborhood Topologies

In the original PSO, two different kinds of neighborhoods were defined for PSO:

In the gbest swarm, all the particles are neighbors of each other; thus, the position of the best overall particle in the swarm is used in the social term of the velocity update equation.

It is assumed that gbest swarms converge fast, as all the particles are attracted simultaneously to the best part of the search space.

In the lbest swarm, only a specific number of particles (neighbor count) can affect the velocity of a given particle. The swarm will converge slower but can locate the global optimum with a greater chance.



Different neighborhoods can be characterized in terms of two factors

•The degree of connectivity, *k*, that measures the number of neighbors of a particle

•The amount of clustering *C*, that measures the number of neighbors of a particle that are also neighbors of each other

The following additional topologies were tested:

•Random
•Von Neumann, a two dimensional grid with neighbors to the N, E, W and S
•Pyramid, a three-dimensional triangular grid
•Star, all the particles connected to a central particle
•Heterogeneous, particles are grouped in several cliques

To know more

THE site:

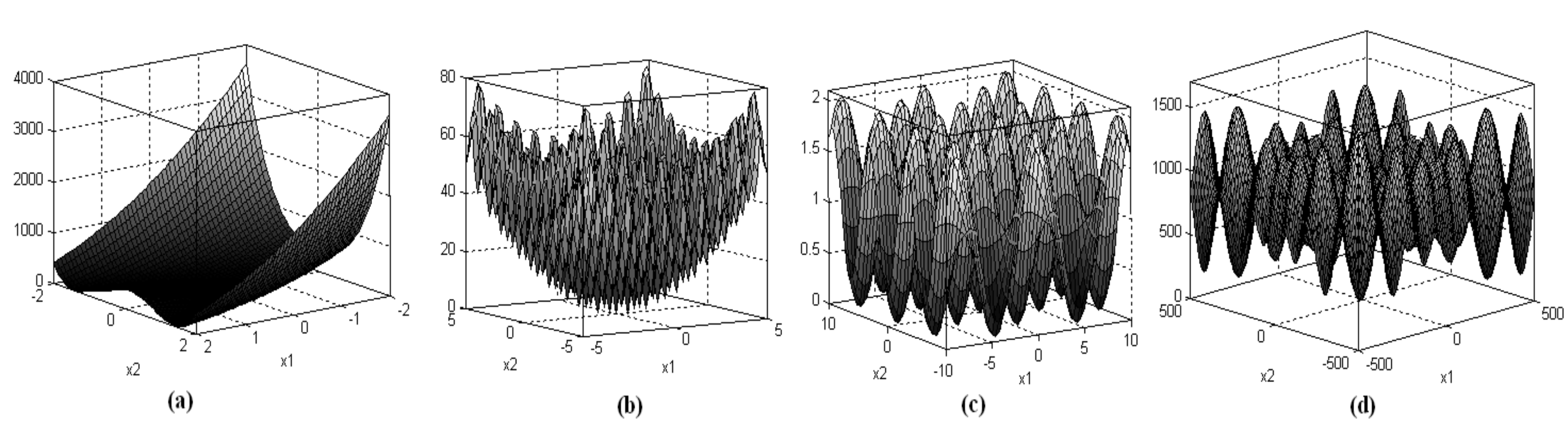
Particle Swarm Central, http://www.particleswarm.net

Clerc M., Kennedy J., "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space", *IEEE Transaction on Evolutionary Computation*, 2002,vol. 6, p. 58-73.

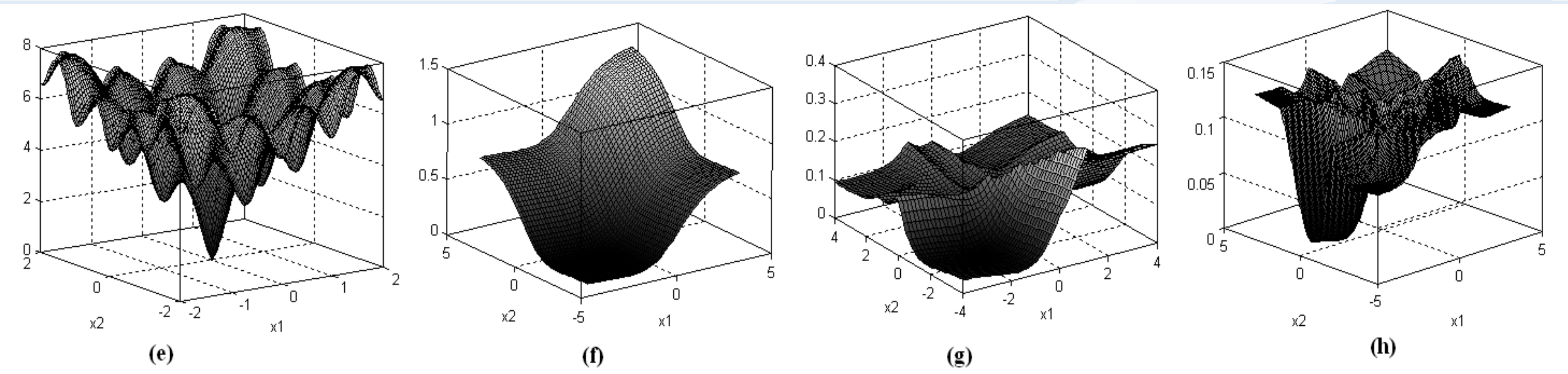
Comparison with other evolutionary computation techniques.

- Unlike in genetic algorithms, evolutionary programming and evolutionary strategies, in PSO, there is no selection operation.
- All particles in PSO are kept as members of the population through the course of the run
- PSO is the only algorithm that does not implement the survival of the fittest.
- No crossover operation in PSO.
- In PSO balance between the global and local search can be adjusted through the inertial weight factor (*w*)

Multi Modal Function



(a) Rosenbrock function, (b) Rastigrin function, (c) Grienwank function, (d) Schwefel function



(e) Ackley function, (f) Kennedy multimodal function generator (M=1 peaks),

(g) Kennedy multimodal function generator (M=10 peaks),

(h) Kennedy multimodal function generator (M=100 peaks).

1. Rosenbrock function

- $f(x) = \sum_{i=1}^{N} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
- Range: $-2 \le x_i \le 2$
- Optimum: Min f = 0.0 for $x_i = 1.0\,, i = 1, \ldots\ldots, N$
- Number of Variables N=500

2. Rastigrin function

- $f(x) = \sum_{i=1}^{N} (x_i^2 - 10\cos(2\pi x_i) + 10)$
- Range: $-5 \le x_i \le 5$
- Optimum: Min f = 0.0 for $x_i = 0.0, i = 1, \ldots, N$
- Number of Variables N=500

3. Griewank function

$$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

- Range: $-10 \le x_i \le 10$
- Optimum: Min f = 0.0 for $x_i = 0.0, i = 1, \dots, N$
- Number of Variables N=500

4. Schwefel function

- $f(x) = 418.9829i + \sum_{i=1}^{N} x_i \sin\left(\sqrt{|x_i|}\right)$
- Range: $-500 \leq x_i \leq 500$
- Optimum: Min f = 0.0 for $x_i = 416.9829, i = 1, \ldots, N$
- Number of Variables N=500

5. Ackley function

- $$f(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos 2\pi x_i\right)$$
- Range: $-2 \le x_i \le 2$
- Optimum: Min *f* = 0.0 for $x_i = 0.0, i = 1, \ldots, N$
- Number of Variables N=500

6. Kennedy multimodal function generator

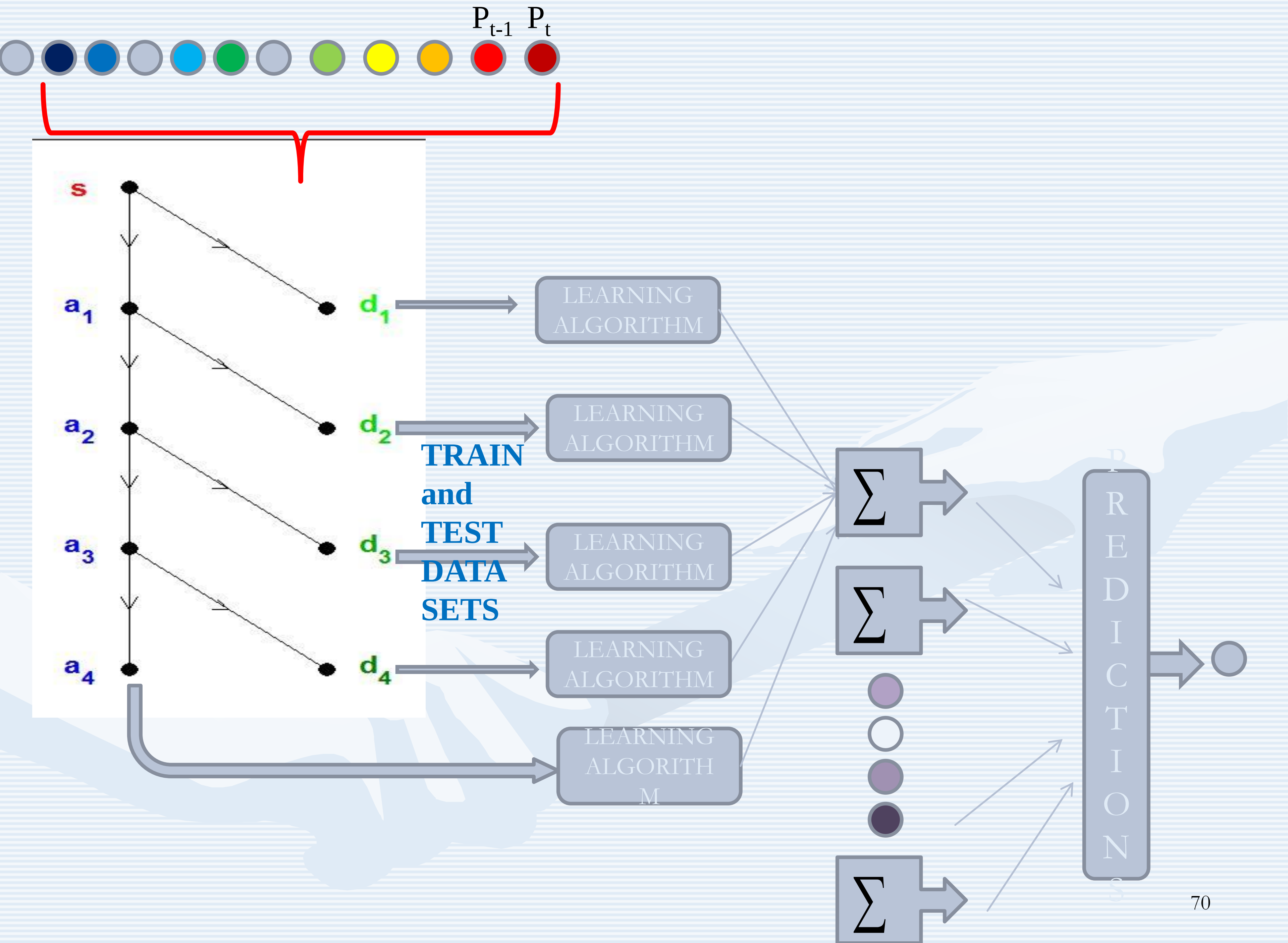
- $$f(x) = \min_{j=1,\ldots,M}\left(\sum_{i=1}^{N}\left[\frac{1}{1+\exp(-x_i)} - A_{ij}\right]^2 + \frac{(j-1)^{0.15}}{15}\right)$$
- Where is the random matrix, i=1,…,N, j=1,…,M
- Range: $-4 \le x_i \le 4$
- Optimum: Min *f* = 0.0 for random $x_i, i = 1, \ldots, N$
- Number of Variables N =5, 10
- Number of Peaks M =10, 100

OPTIMUM RESULTS AFTER 1000 INDEPENDENT SIMULATION RUNS

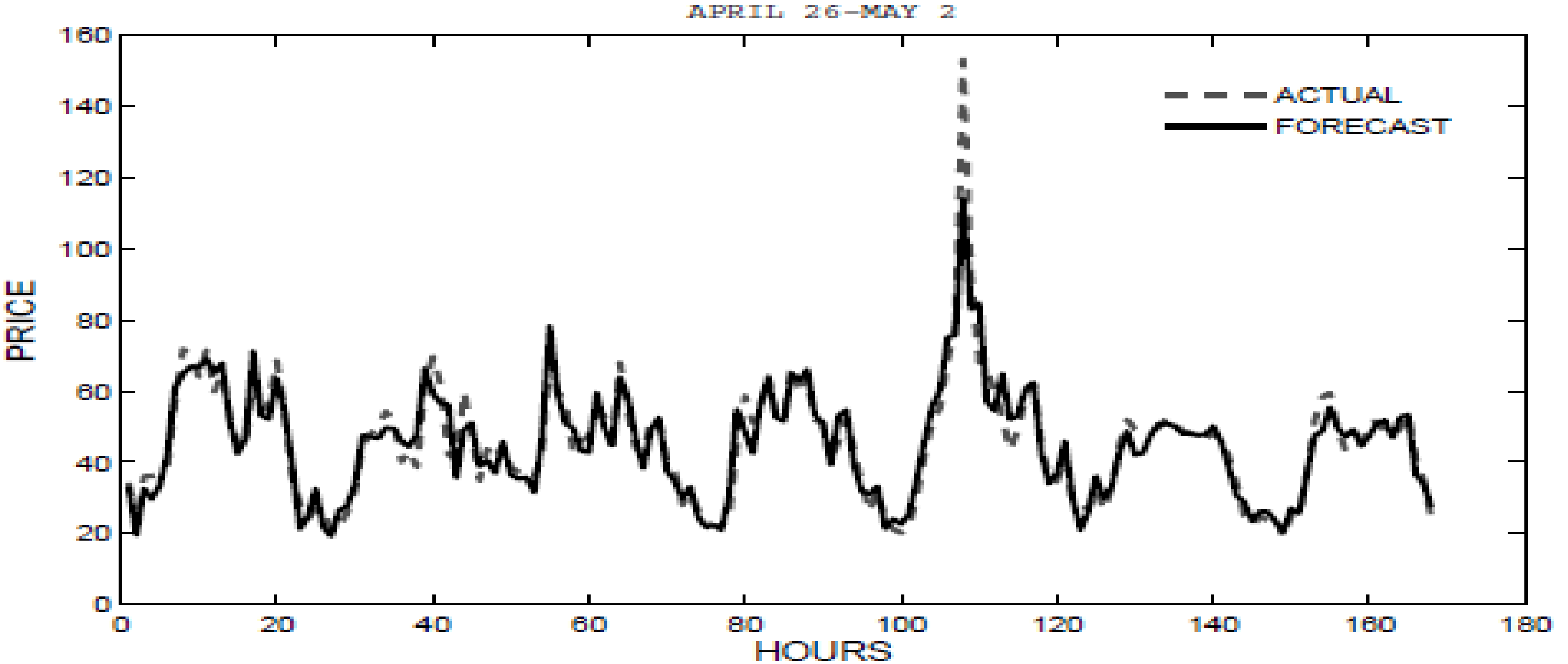
Test Functions	PSO	
	Avg.	Std. Dev.
Rosenbrock (N=500)	3382.72	480.7856
Rastigrin (N=500)	5626.28	278.9639
Griewank (N=500)	1.214795	0.045499
Schwefel (N=500)	144703.83	12418.28
Ackley (N=500)	2.503970	0.186395
Kennedy (N=5,M=100)	0.000297	0.000000
Kennedy (N=10,M=10)	0.003017	0.000000
Kennedy(N=10,M=100)	0.122232	0.000000

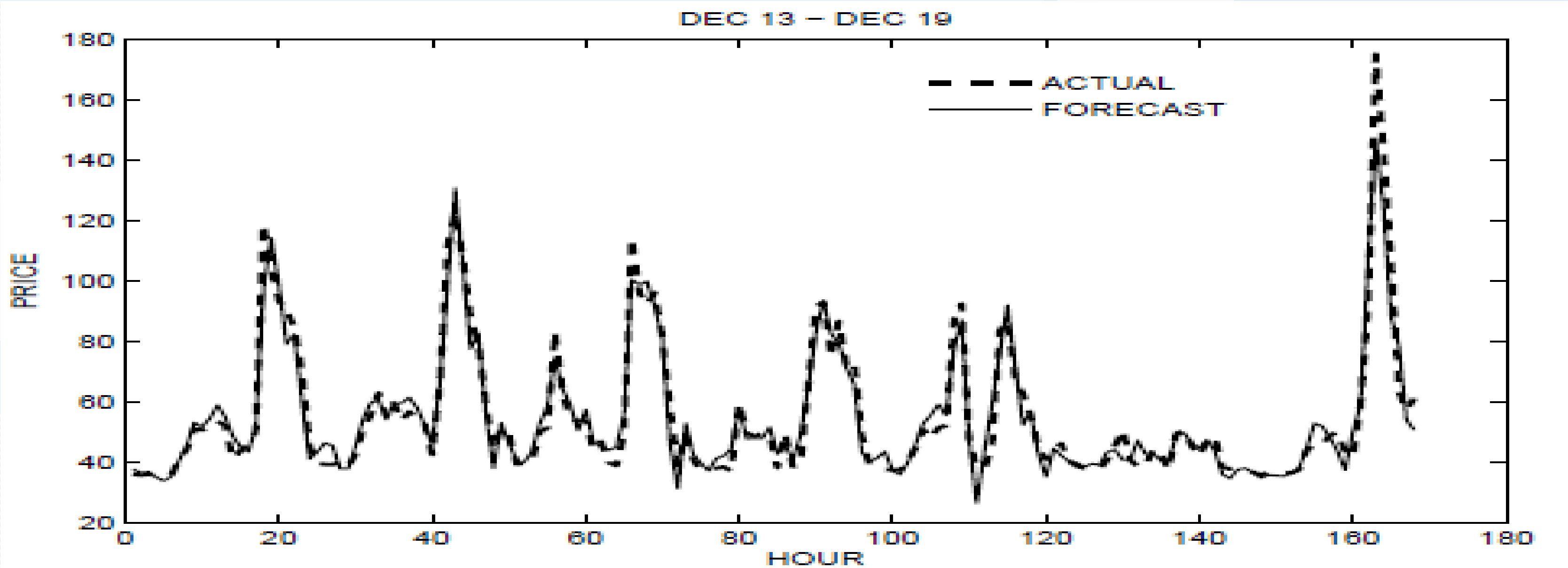
Electricity Price Forecasting

Hybrid Tools and Techniques



P_{t-1} P_t
s
a_1
a_2
a_3
a_4
d_1
d_2
d_3
d_4
TRAIN and TEST DATA SETS
LEARNING ALGORITHM
LEARNING ALGORITHM
LEARNING ALGORITHM
LEARNING ALGORITHM
LEARNING ALGORITHM
∑
∑
∑
PREDICTIONS


APRIL 26–MAY 2
ACTUAL
FORECAST
PRICE
HOURS
0
20
40
60
80
100
120
140
160
180


DEC 13 – DEC 19
ACTUAL
FORECAST
PRICE
HOUR
0
20
40
60
80
100
120
140
160
180

Publications based on research work

[1] Nitin Anand Shrivastava, Abbas Khosravi and Bijaya Ketan Panigarhi, "Prediction interval estimation for electricity prices using PSO tuned support vector machines," *IEEE Transactions on Industrial Informatics,* vol. 11, no.2, pp. 322-331, 2015.

[2] Nitin Anand Shrivastava and Bijaya Ketan Panigarhi, "Prediction intervals for electricity price and demands: A Multi-Objective approach," *IET Generation Transmission and Distribution*, vol. 9, no. 5, pp. 494–502, 2015.

[3] Nitin Anand Shrivastava and Bijaya Ketan Panigarhi, "A hybrid wavelet-ELM based short term price forecasting for electricity markets," *International Journal of Electrical Power and Energy Systems*, Elsevier, Volume 55, February 2014, Pages 41-50.

[4] Nitin Anand Shrivastava and Bijaya Ketan Panigarhi, Meng-Hiot Lim, "Electricity price classification using extreme learning machines," *Neural computation and applications*, Springer-Verlag London 2014, DOI 10.1007/s00521-013- 1537-1.

[5] Nitin Anand Shrivastava and Bijaya Ketan Panigarhi, "Point and prediction interval estimation for electricity markets with machine learning techniques and wavelet transforms," *Neurocomputing*, Elsevier, Volume 118, October 2013, Pages 301-310.

Publications based on research work

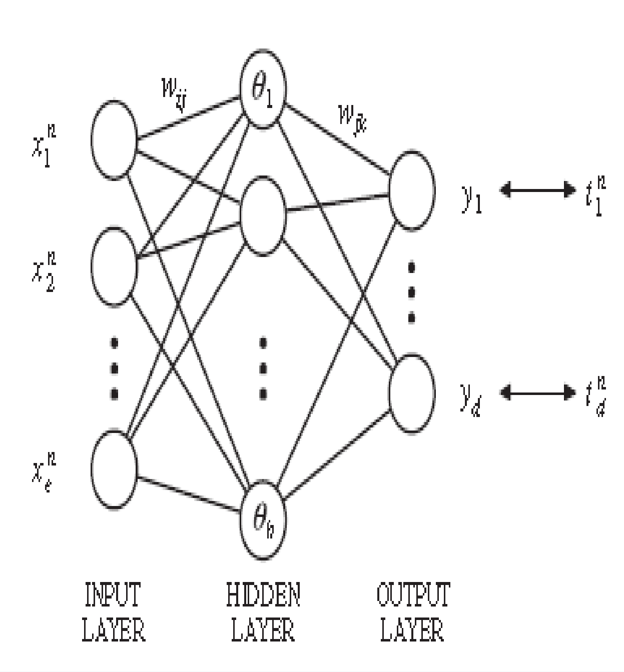
[6] Nitin Anand Shrivastava and Bijaya Ketan Panigarhi, "A hybrid swarm-machine itelligence approach for day ahead price forecasting." *Int. J. Innov. Comput. Appl.*, Inderscience, Volume 3, 4, December 2011, Pages 177-187.

[7] Nitin Anand Shrivastava, Bijaya Ketan Panigrahi and Meng-Hiot Lim, "Electricity price classification using extreme learning machines," *International Conference on Extreme Learning Machines (ELM2013)*, Beijing, China, October 15-17, 2013.

[8] Nitin Anand Shrivastava, Abbas Khosravi, Bijaya Ketan Panigrahi, Prediction interval estimation for electricity price and demand using support vector machines, IJCNN 2014, Beijing, China, pp. 3995-4002.

Uncertainty and Forecasting

- Traditionally, neural networks used for prediction purposes give rise to a point prediction when they are presented with a set of input values. However, there is always a degree of uncertainty associated with any point prediction.
- That uncertainty is attributable to either structure of the model or the inherent uncertainty in the data set used for model development.
- Due to these reasons, point prediction performance deteriorates and predictions become unreliable.



Uncertainty and Forecasting

- The reliability of point forecasts significantly drops due to the prevalence of uncertainty in operation of the system.
- Even if uncertainties are known or predictable, the targets will be multi-valued, making predictions prone to error.
- This weakness is due to the theoretical point that NNs generate averaged values of targets conditioned on inputs.
- Such a reduction cannot be mitigated through changing the model structure or repeating the training process
- NN provide point predictions without any indication of their accuracy.
- Point predictions are less reliable and accurate if the training data is sparse, if targets are multi-valued, or if targets are affected by probabilistic events.

Need for incorporating uncertainties

- When forecast results are presented to end users, they should be informed as **to what extent they can be trusted**.
- Availability of prediction intervals will allow the decision makers to efficiently **quantify the levels of uncertainties** associated with the point forecasts, and to consider **a multiple of solutions for different conditions**.

Confidence Intervals and *Prediction Intervals*

- Targets can be modeled as

$$t(x)=f(x)+e(x)$$

$t(x)$ → *observed target value*

$f(x)$ → *true regression*

$e(x)$ → *noise with zero mean*

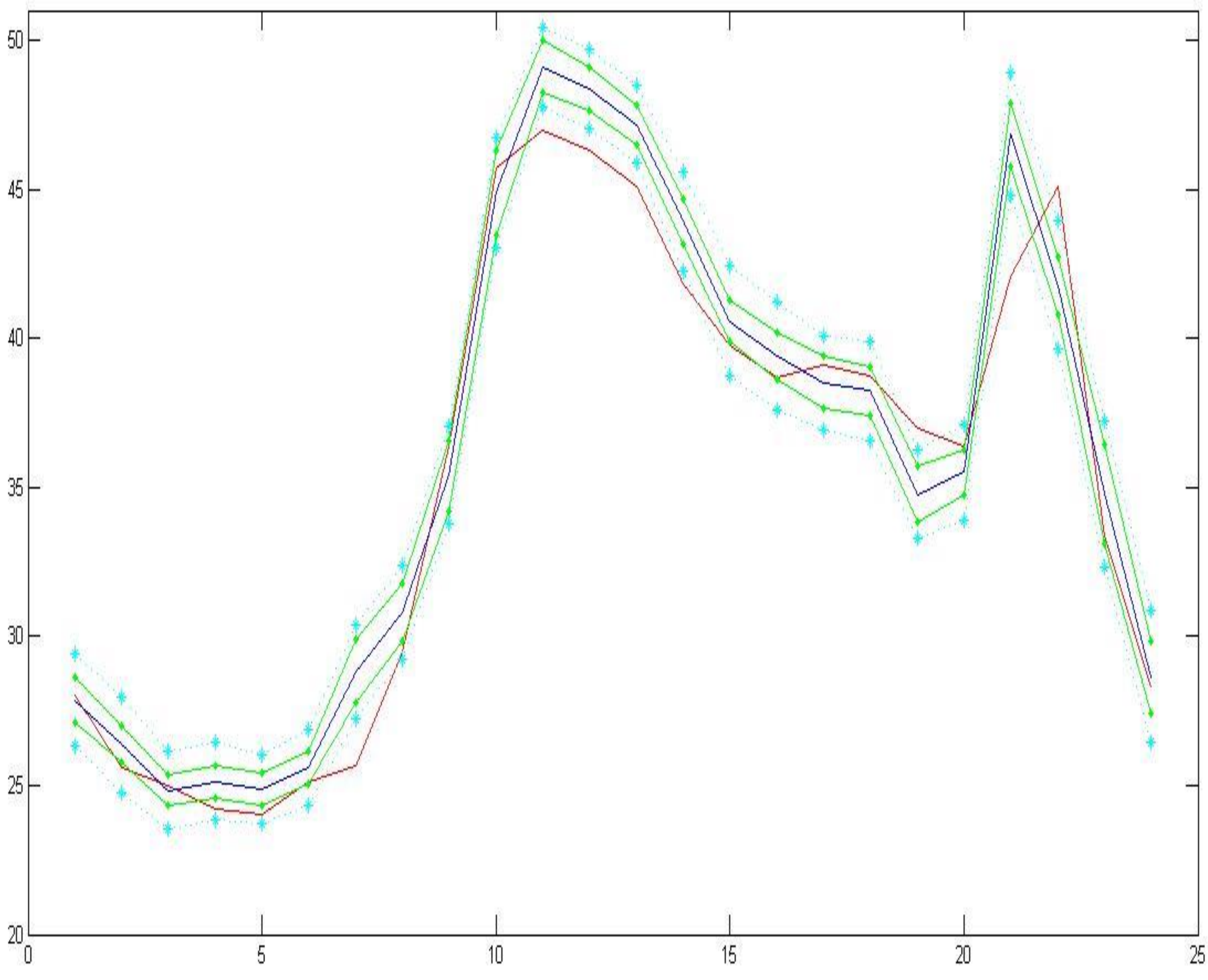
Training a ML algorithm is meant to estimate $\varphi(x)$ i.e. an approximation of $f(x)$. It is an estimation of the mean of the distribution of the target values given an input vector x.

Confidence Intervals and *Prediction Intervals*

- Two measures of the confidence of this point prediction

1. Confidence intervals: accuracy of our estimate of true regression i.e. distribution of quantity ***f(x)- φ(x)***
2. Prediction intervals: estimate of confidence in prediction of targets themselves i.e. distribution of quantity ***t(x)- φ(x)***

- $t(x) - \varphi(x) = [f(x) - \varphi(x)] + e(x)$


50
45
40
35
30
25
20
0
5
10
15
20
25

Uncertainty estimating techniques

- ***Delta technique:*** Based on the understanding that multilayer feedforward neural networks are basically non-linear regression models which can be linearized using Taylor's series expansion .
- ***Bayesian technique :*** Each parameter in a neural network is considered as a distribution rather than a single value and therefore the network outcomes will also be in the form of distributions conditional on the observed training set . It suffers from the limitation of massive computational burden and calculation of Hessian matrix.
- ***Bootstrap method:*** Bootstrapping is a computer based statistical technique proposed by which is meant to approximate the unknown probability distribution of an estimator or statistic by an empirical distribution obtained by a resampling process.

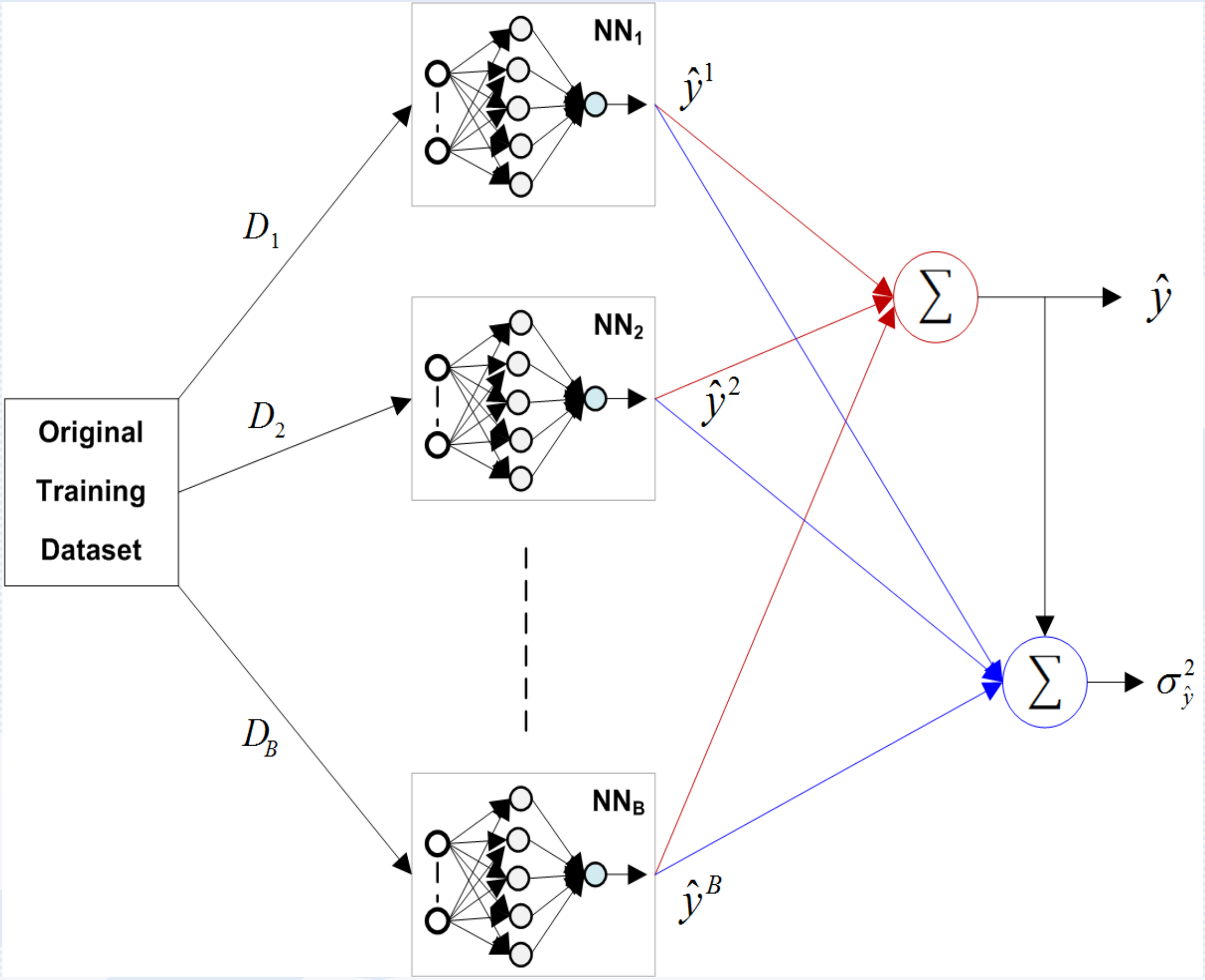
Bootstrapping technique

- 1. Efficient and easy to use in comparison to Bayesian method involving complex computations.
- 2. It gives rise to input-dependent values for variance in comparison to the delta method which makes a strong case for its relevance in constructing reliable prediction intervals.
- 3.Non-parametric alternative to conventional methods and are designed to produce asymptotically correct coverage rates under no specific assumption about the error distribution.

BOOTSTRAP METHOD FOR PI CONSTRUCTION

- Fusion of multiple estimators/models to improve the overall prediction performance
- *B training datasets* are resampled from the original dataset with replacement.
- *The method estimates the variance due* to model mis-specification, $\sigma_{\hat{y}}^2$, *by building B NN_y models*
- The true regression is estimated by averaging the point forecasts of *B models*, $\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B}\hat{y}_i^b$

Bootstrap technique



Methodology

- Assuming that NN models are unbiased, the model mis-specification variance can be estimated using the variance of B *model outcomes,*

$$\sigma^2_{\hat{y}_i} = \frac{1}{B-1}\sum_{b=1}^{B}(\hat{y}_i^b - \hat{y}_i)^2$$

- To construct PIs, we also need to estimate the variance of errors, $\sigma^2_{\hat{\varepsilon}_i}$. *The key idea is to develop a separate individual* NN model, called NN_{σ_ε} *to provide an estimate of* $\sigma^2_{\hat{\varepsilon}_i}$ *when* presented with an input vector.

PI ASSESSMENT MEASURES

- PI Coverage Probability (PICP) is measured by counting the number of target values covered by the constructed Pi:

$$PICP = \frac{1}{n}\sum_{i=1}^{n} c_i$$

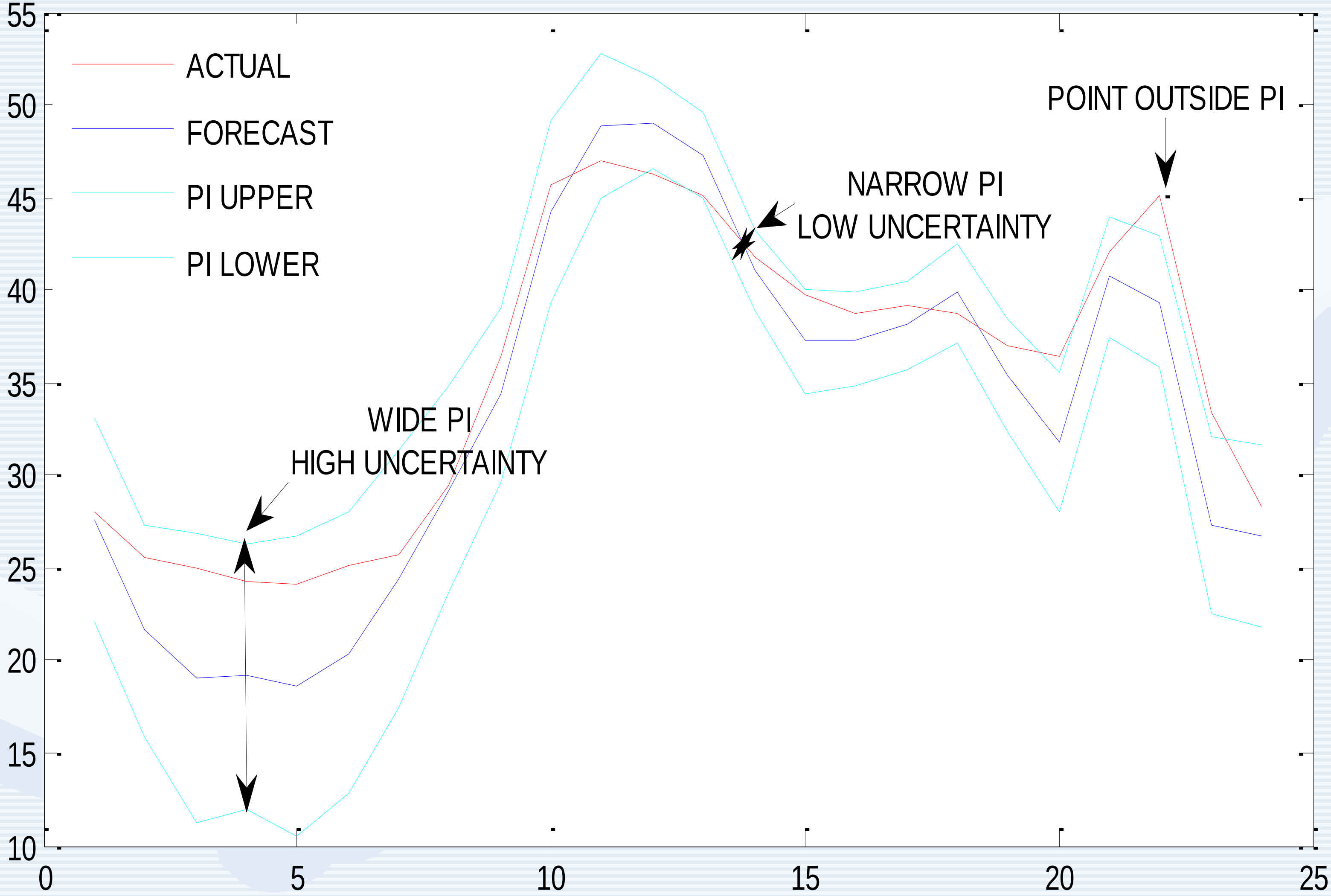
Where

$$c_i = \begin{cases} 1 & t_i \in [L_i, U_i] \\ 0 & t_i \notin [L_i, U_i] \end{cases}$$

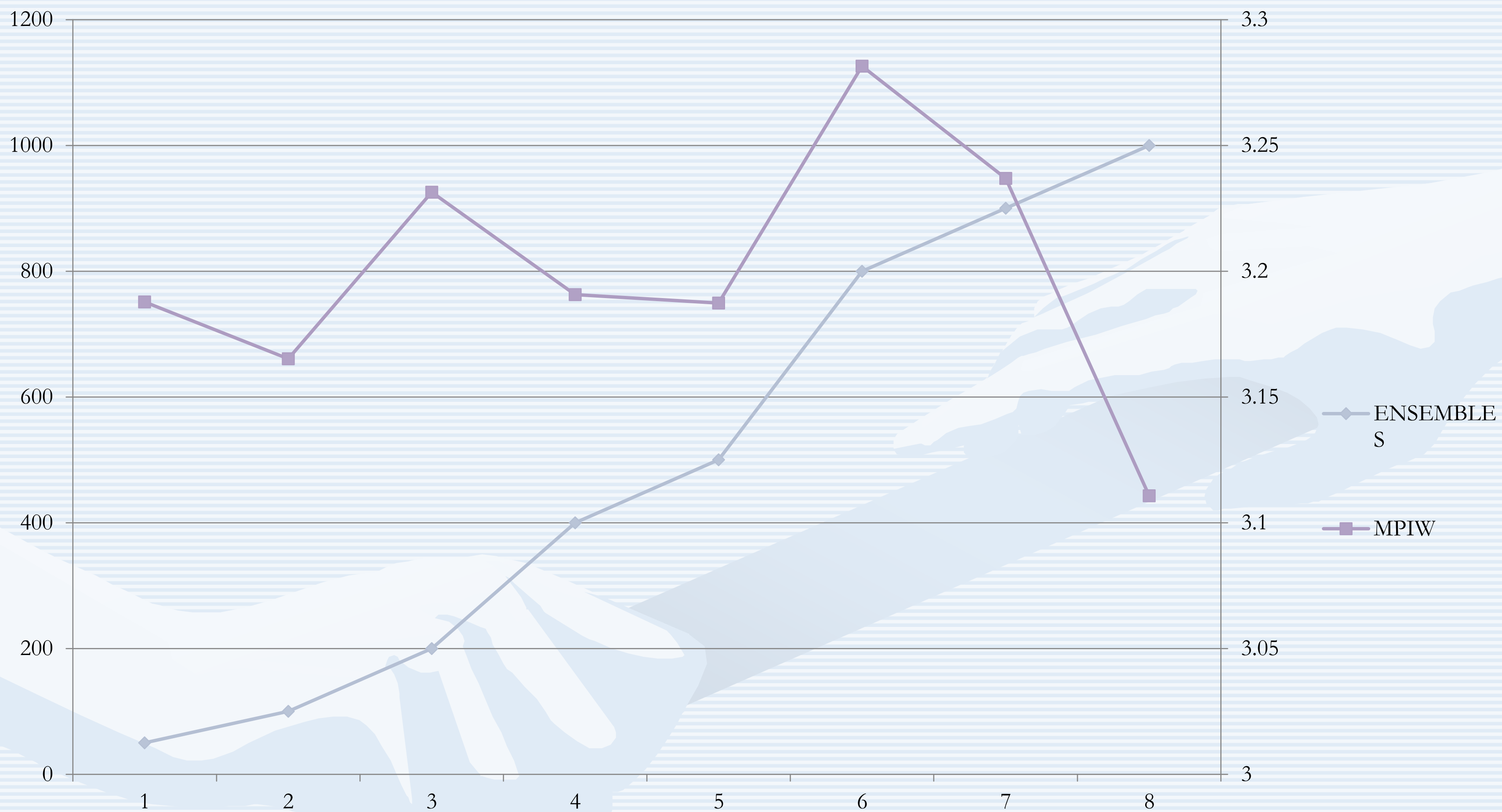
- It is essentia based on their width. Mean PI Wi uantifies how wide constructed PIs are,

$$MPIW = \frac{1}{n}\sum_{i=1}^{n}(U_i - L_i)$$

Inferences from PI



Experimental Results



EXPERIMENTAL RESULTS

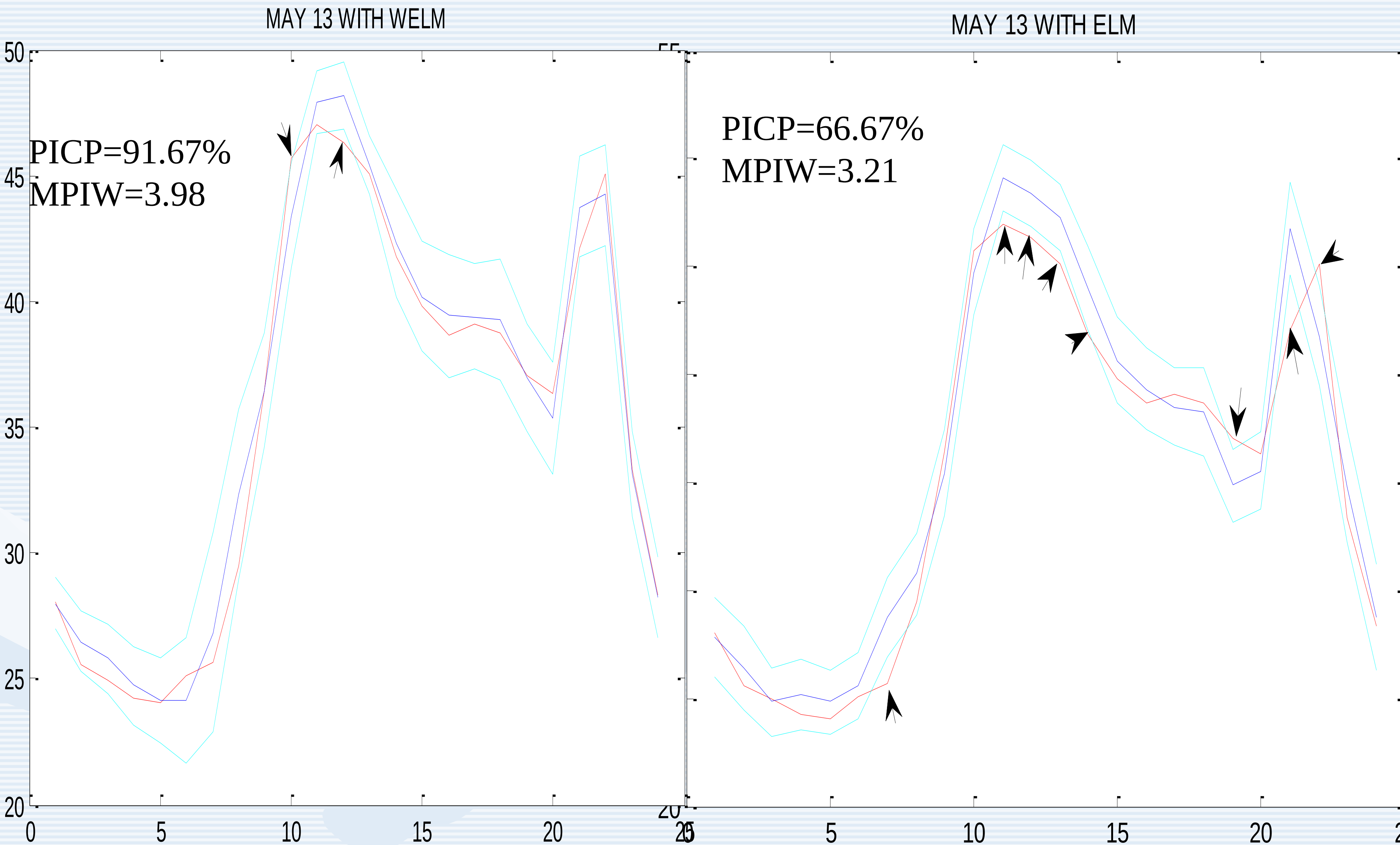
TEST DAY	ANN			ELM		
	MAPE	PICP(%)	MPIW	MAPE	PICP(%)	MPIW
JAN 20	7.28	100	22.51	3.31	79.17	5.60
FEB 10	5.03	100	23.98	2.64	87.50	6.20
MAR 5	7.35	87.50	15.71	2.14	95.83	4.86
APR 7	9.84	95.83	26.15	4.12	83.33	8.56
MAY 13	8.59	83.33	9.06	2.33	91.67	3.98

Results for Basic ELM model with different hidden nodes.

NODES	10	20	30	40	50	60	70	80	90
MAPE	5.55	4.46	4.10	3.89	3.68	3.58	3.48	3.42	3.37
PICP	79.17	62.50	62.50	66.67	66.67	66.67	66.67	66.67	66.67
MPIW	6.56	3.64	3.28	3.23	3.26	3.25	3.37	3.29	3.31

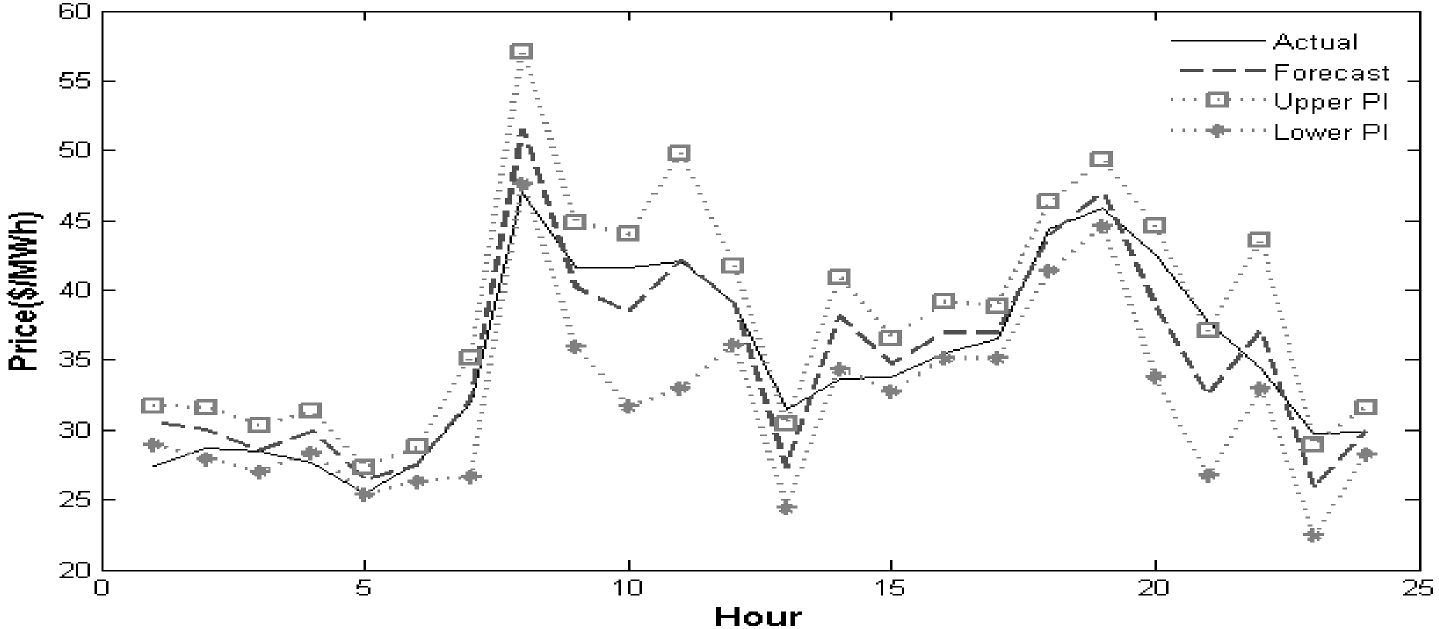
Results for Basic ELM model with different number of Ensembles.

ENSEMBLES	10	50	100	200
MAPE	3.71	3.35	3.38	3.41
PICP(%)	79.17	79.17	83.33	83.33
MPIW	4.53	4.28	4.60	4.55
TIME(Sec)	188	846	1647	3325

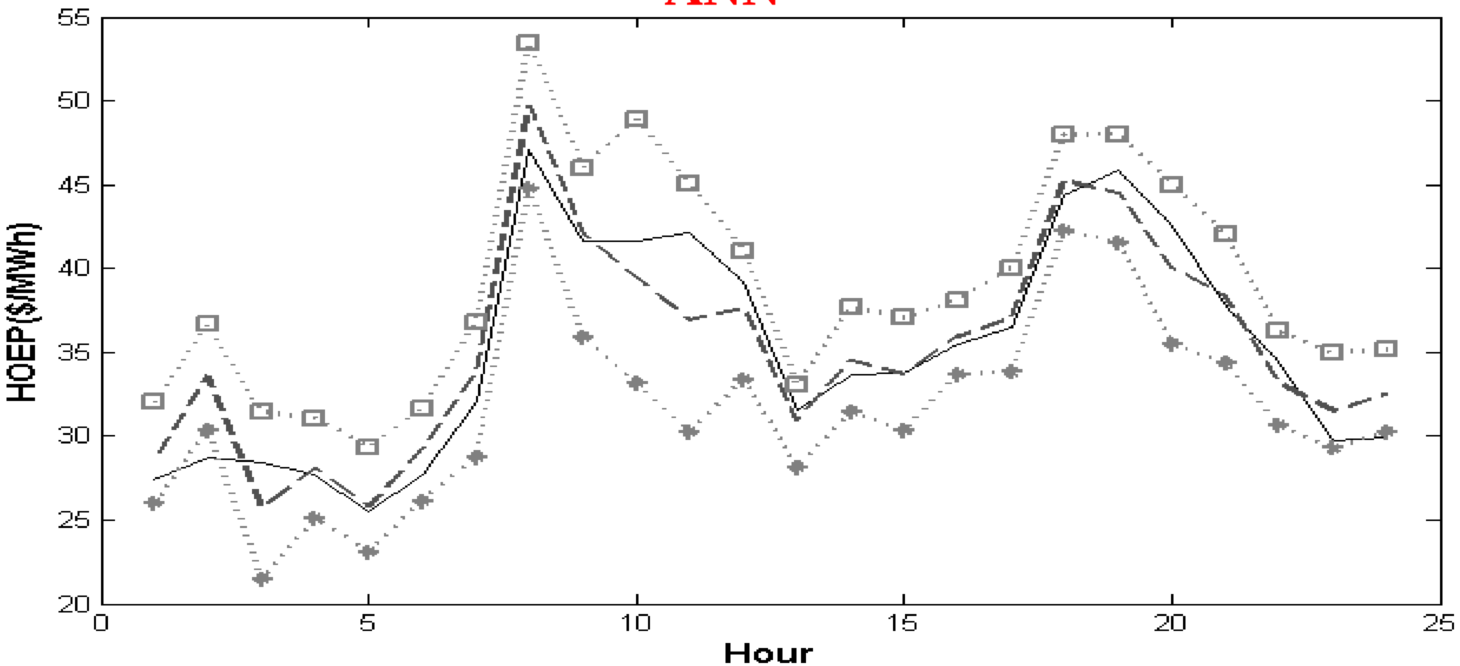

MAY 13 WITH WELM
PICP=91.67%
MPIW=3.98
MAY 13 WITH ELM
PICP=66.67%
MPIW=3.21

ONTARIO MARKET

ELM

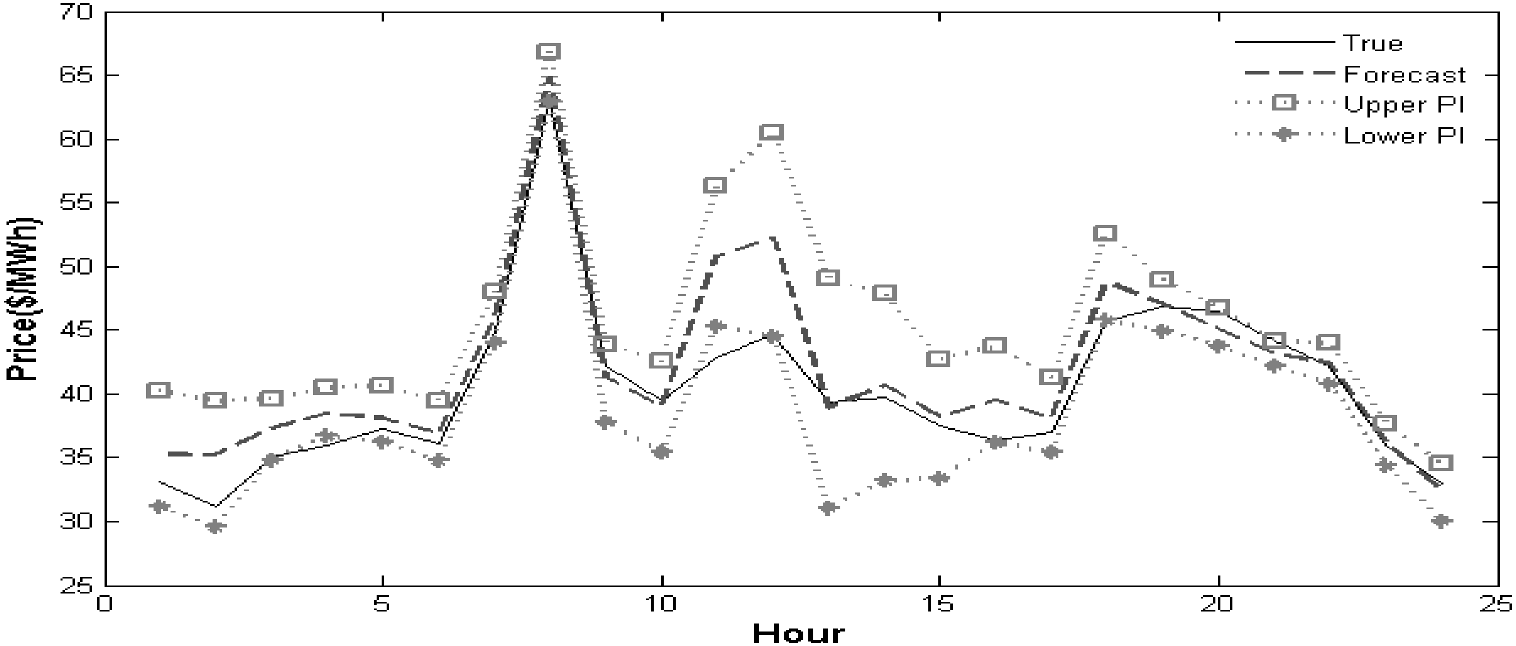


ANN

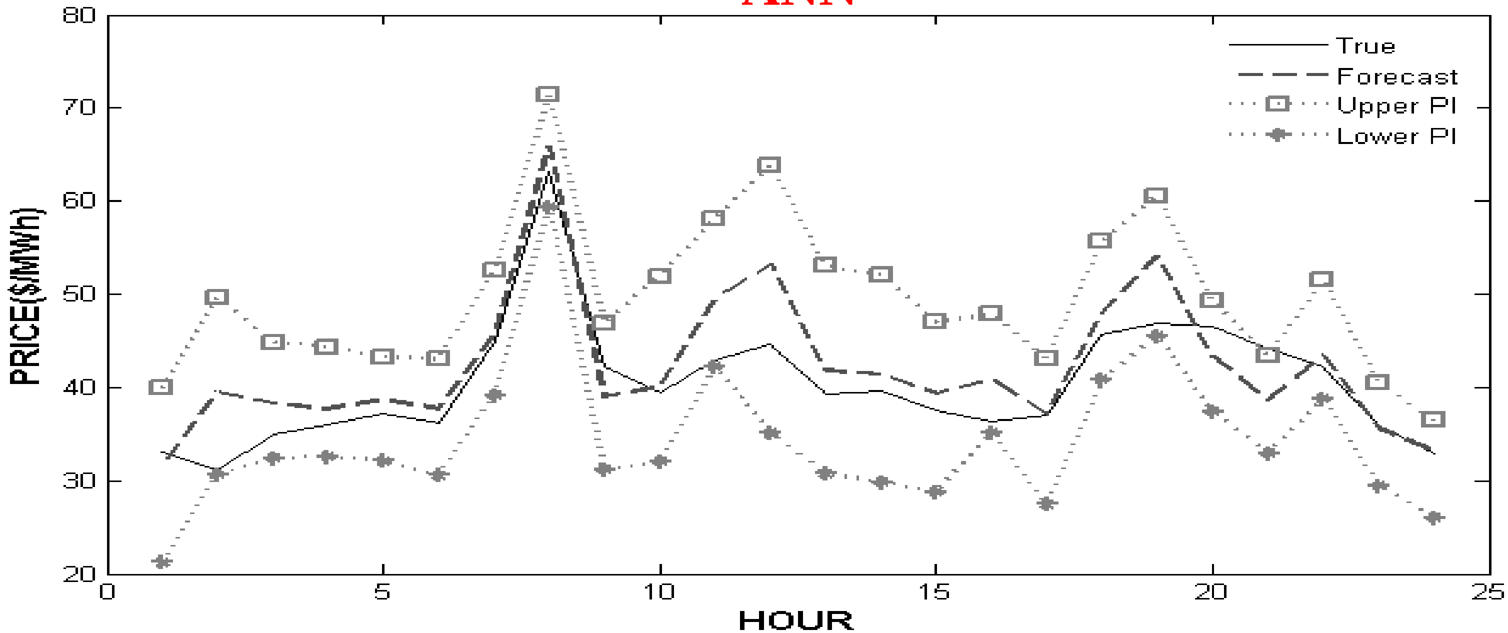


JM REAL-TIME MARKET

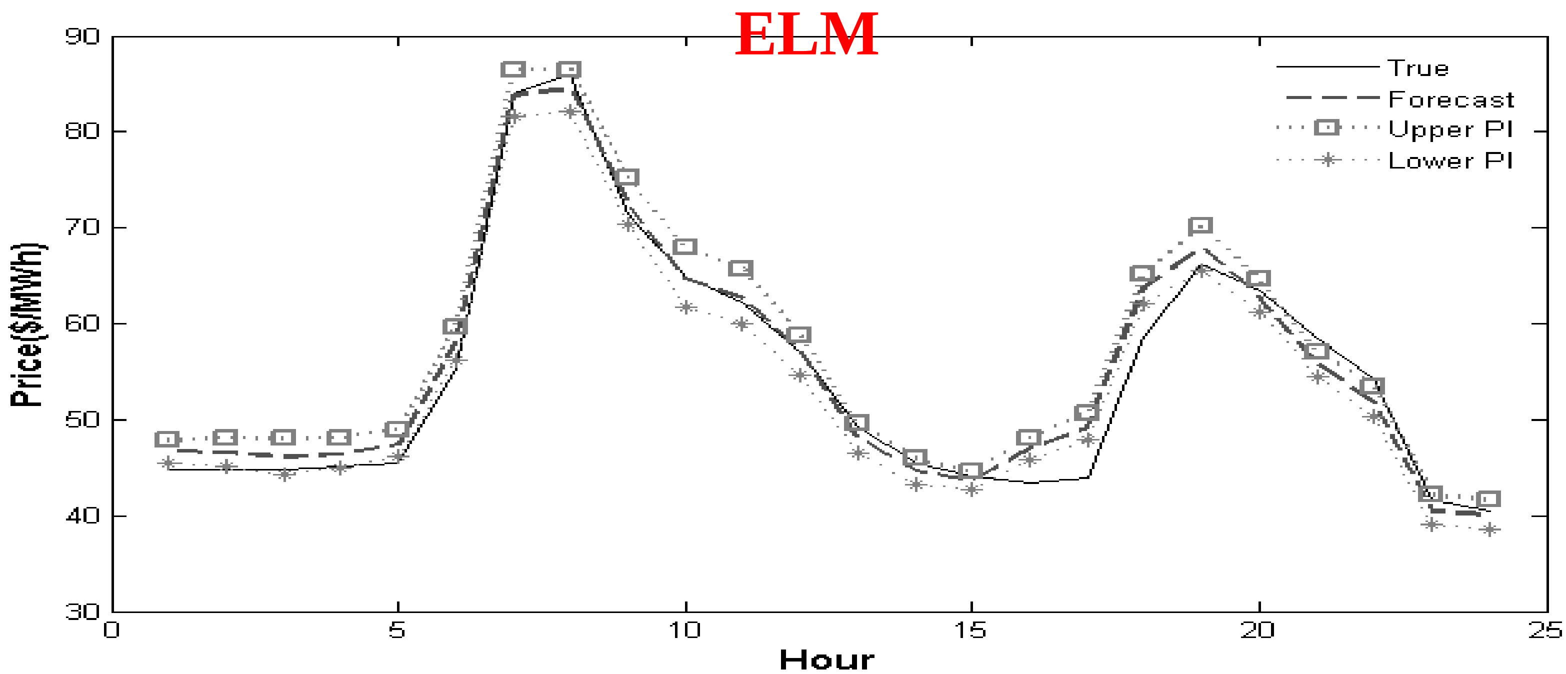
ELM

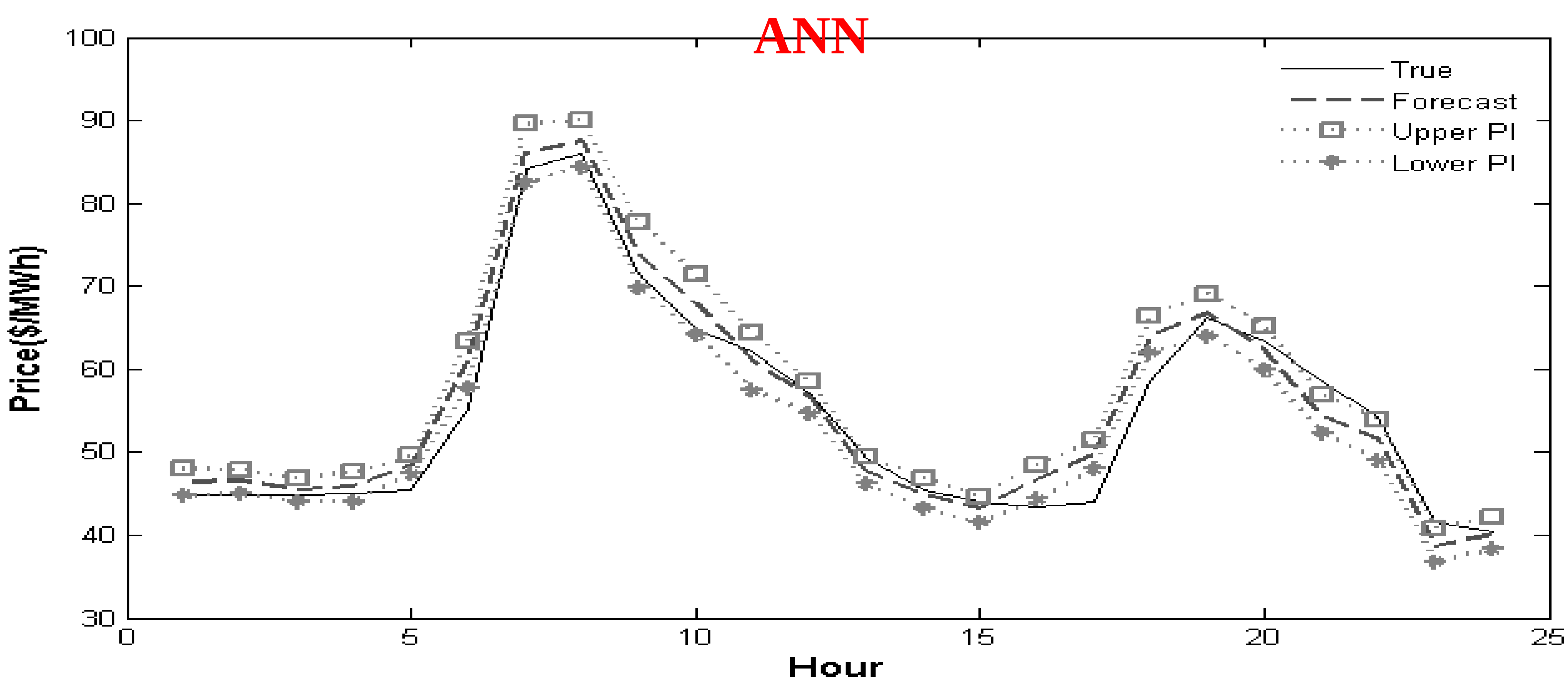


ANN



JM DAY AHEAD MARKET





Biometric Based Personal Authentication: An Introduction

Introduction : The need to Identify

- Every day we are required to authenticate ourselves
  - Using a bank card with a PIN at a cash machine
  - A password to log on to a computer
  - Using a key to open a door
  - Providing a passport and driving licence as proof of identity
- We need to be able to accurately IDENTIFY an individual to minimize current issues and threats

Introduction: Is Biometrics The Answer

- A biometric is part of the person and is not easily compromised through:
  - Theft
  - Collusion
  - Loss
- Simplifies user management resulting in cost savings
- Users do not need to remember passwords
- Users do not need to remember PINs
- Easy to use

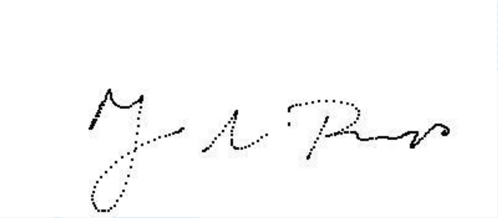
Introduction: Biometric Definition

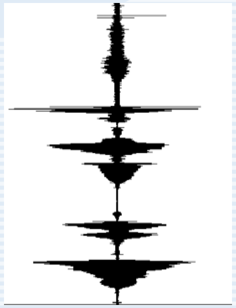
- Biometrics Definition [International Biometrics Industry Association (IBIA)]

- "Biometrics technology involves automatic identification or verification of an individual based on physiological or behavioral characteristics."

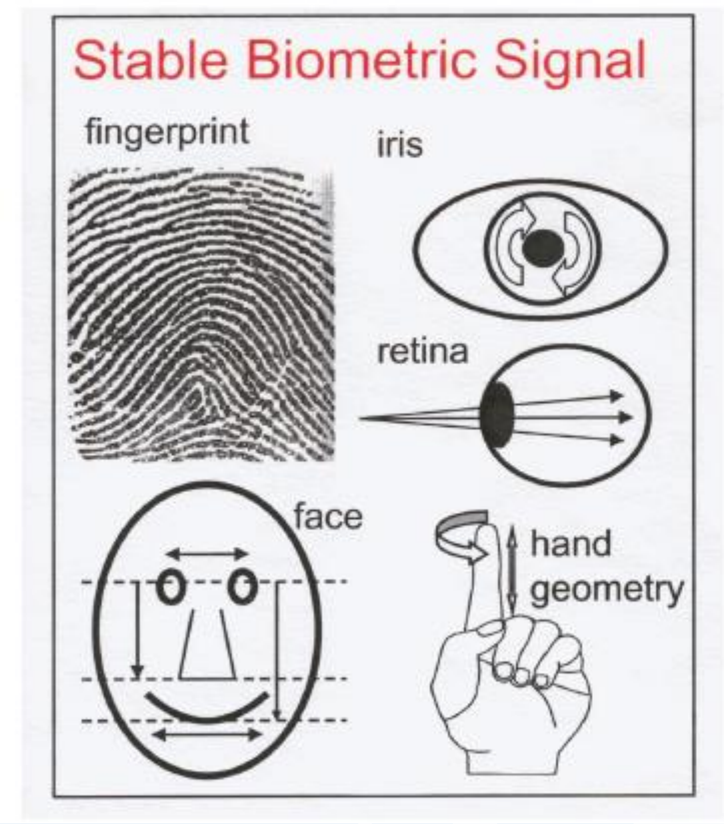
Introduction: Physiological and Behavioural

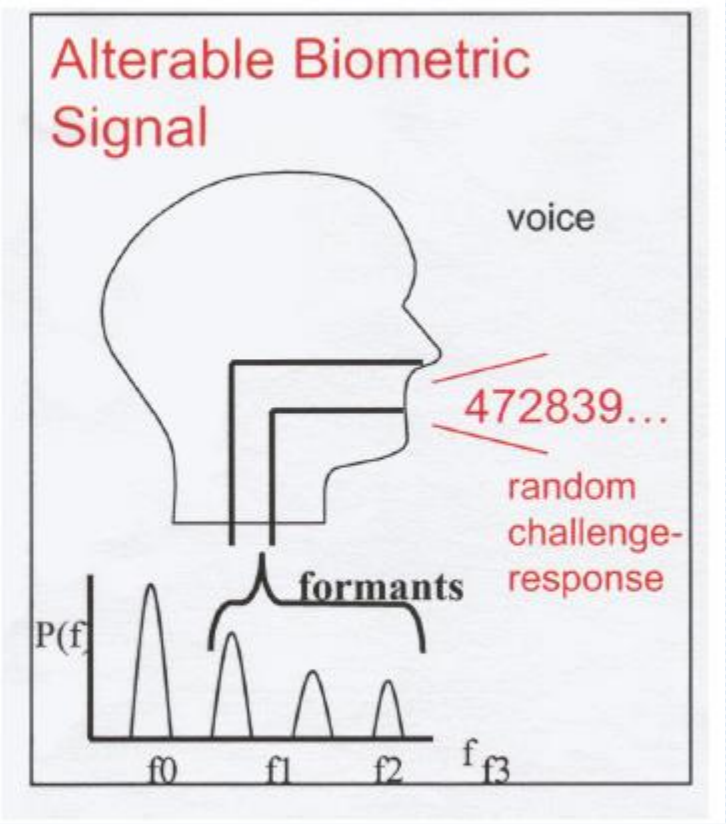
- Physiological
  - Fingerprint
  - Iris
  - Vein pattern
  - Hand geometry
  - DNA
- Behavioural
  - Signature
  - Gait
  - Voice
  - Keystroke dynamics





Introduction : Classes of biometrics





Current Problems (Motivations)

- Iris-Based Biometrics
  - Greatly affected by eye diseases like: cataract, viral infection
  - Acquisition itself a challenge
- Ear Based Biometrics
  - Not a very potential biometrics
  - Effective in combination with other biometric modalities
  - Very less incorporated for multimodal Biometrics
- Multimodal System
  - No efforts for combining uniqueness of face, iris and ear for multimodal systems
  - Selection of effective fusion Scheme
  - Simultaneous image acquisition for online verification itself is very challenging task.

Current Problems (Motivations)

- Face-Based Biometric Systems
  - Digital Faces
    - Pose variations
    - Emotion Variation
    - Light Illumination Problem
    - Disguise
    - Identical Twins
  - Thermal Faces
    - Variation in facial thermograms due to change in Ambient temperature
    - Little/Negligible effort for online applications
    - Very less utilized in combination to other biometric modalities

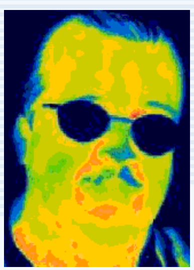
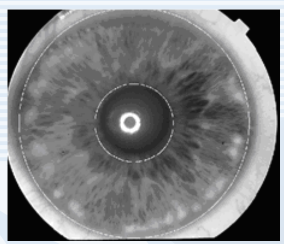
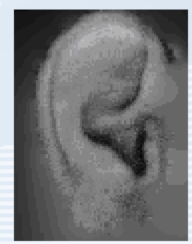
Objectives

- Face Authentication Using Infrared Thermal Imaging
  - To reduce many problems associated with digital imaging in face recognition

Facial thermograms: Stable at Pose Variations

Objectives

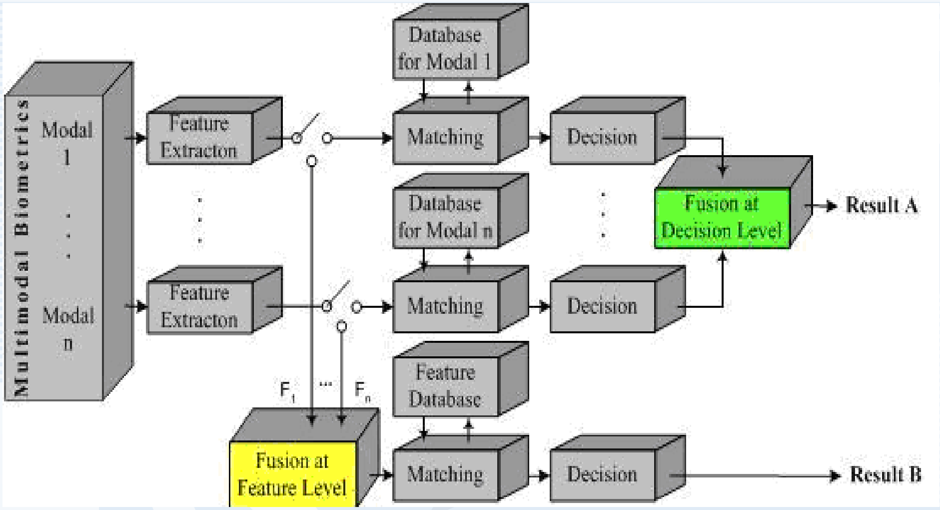
- Investigation of Ear Based Authentication
  - To reduce the shortcomings of the individual Biometrics and achieve the better security standards
  - To investigate a novel multimodal (Face-based) biometric system based on Thermal faces, Irises, and Ears; never employed so far for online applications.

 $\otimes$  $\otimes$ 

Multimodal System

Objectives

- An effective Fusion Scheme for Integration
  - Feature Level Fusion Scheme for integration of proposed modality
  - Decision Level Fusion Scheme as alternative to choose a better and relevant fusion scheme based on performance index.

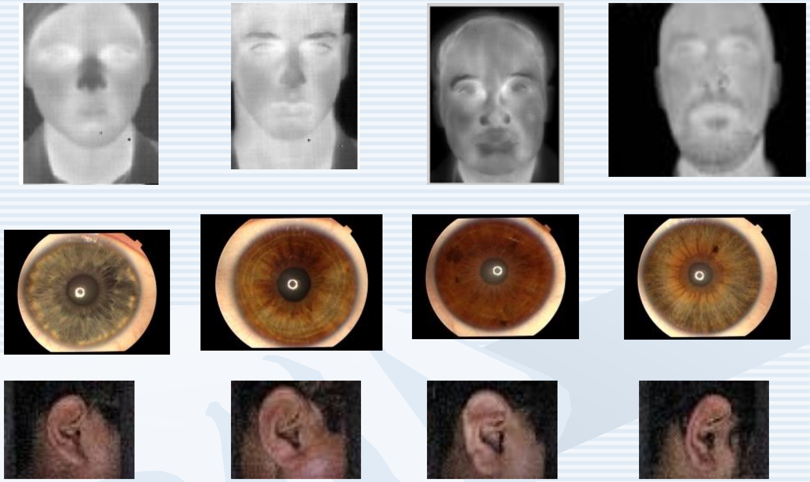


The General Framework for Fusion of Multimodal Biometrics

Objectives

- A Multimodal Database

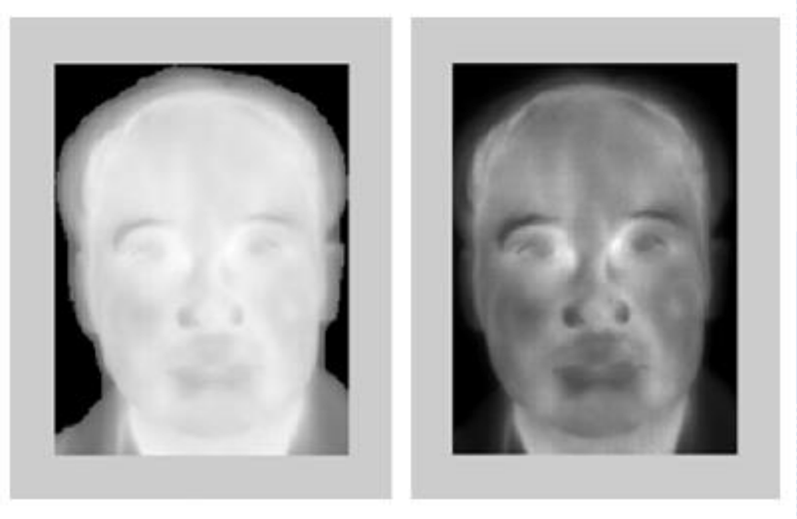
  - Building a Multimodal Database containing 2000 Thermal faces, Irises, and Ears of 200 individuals (10 images each)
  - To investigate the proposed approaches on the created database; not available so far for research purposes.



A Sample Multimodal Database

Proposed Approaches (Face Recognition)

- Skin Heat Transfer (SKT) based facial thermograms
  - To tackle ambient temperature based within-class scatter of facial images

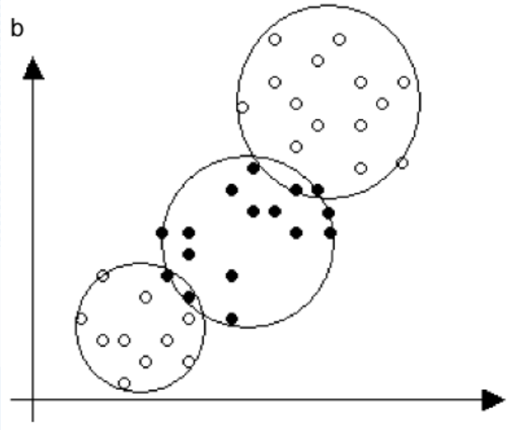


(a) Thermal data *(Te=26.2 °C)* (b) Corresponding SKT data.

Proposed Approaches (Face Recognition)

- Radial Basis function (RBF) Neural networks based supervised clustering technique for classification

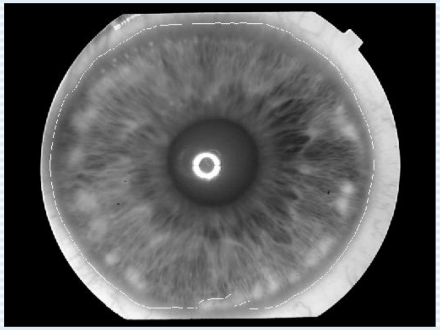
    - To overcome the Linear subspace assumption by traditional methods like Fisher face approach and which is violated by Face data that contain much more overlapping



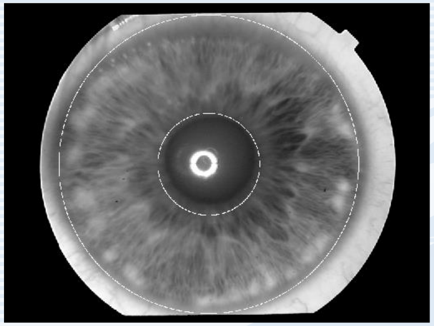
Supervised clustering with homogeneous samples

Proposed Approaches (Iris Recognition)

- Iris Authentication
  - Localization of Outer and Inner Boundary of Iris to normalize images and reduce several acquisition problems



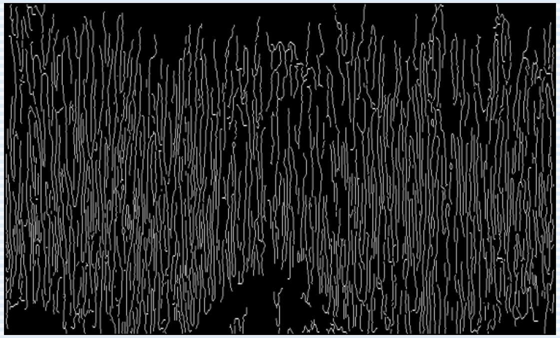
(a) Iris sample image

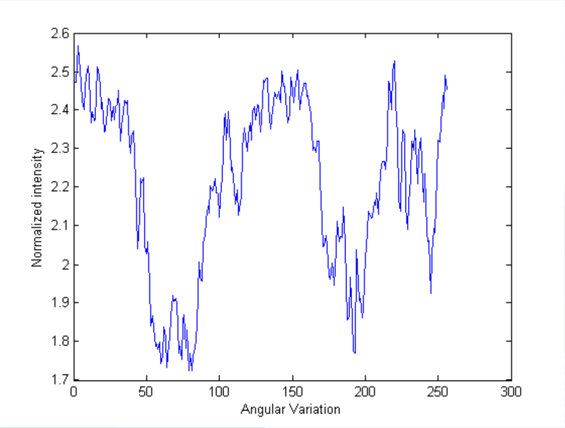


(b) Image after fitting concentric circles

Proposed Approaches (Iris Recognition)

- DWT and Gabor based Features
  - Both Feature representation mechanism is utilized for final authentication



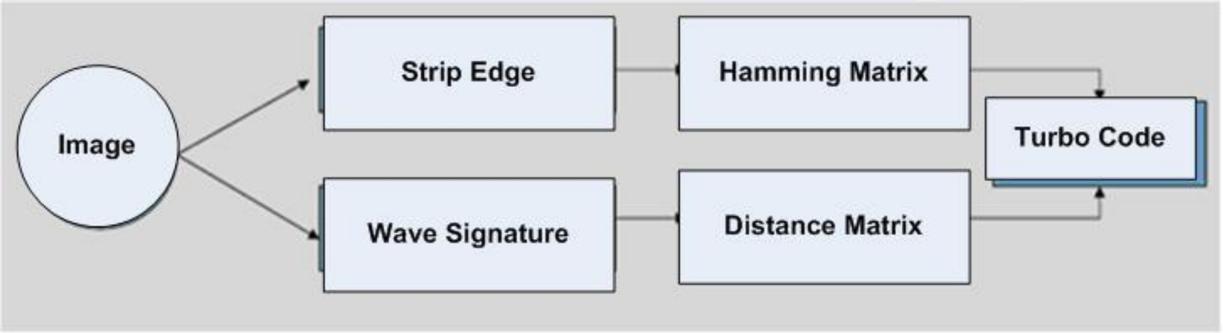


(a) Gabor Based Strip Edge (b) DWT Based Wave-Signature

Department of Electrical Engineering, IIT Delhi, New Delhi, India

Proposed Approaches (Iris Recognition)

- Euclidean and Hamming Distance based Classifiers
  - Matching Scores for each classifiers are computed separately.
  - Turbo code based mechanism for final concatenation of Scores



Block Diagram representing the Turbo Code analogy

Proposed Approaches (Ear Recognition)

- ROI Segmentation

  - Level Set Formulation technique for segmenting ear from image.
  - Laplacian and Canny Based Edge detection

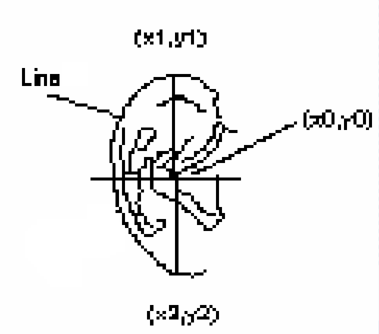
Level Set   Segmented Ear   Ear Edge

Department of Electrical Engineering, IIT Delhi, New Delhi, India

Proposed Approaches (Ear Recognition)

- Texture and Geometrical Features

  - Concentric circle method for distance and angle features .
  - Least Square curve approximation based shape features
  - Gray values based Textural Features



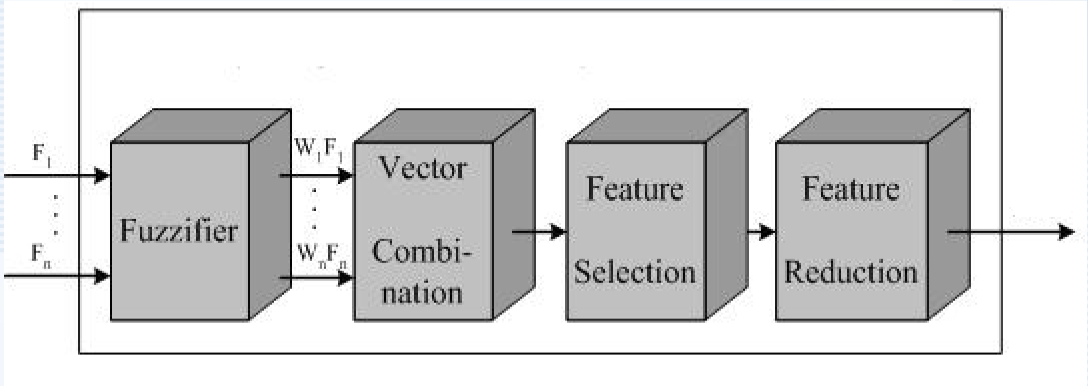
Curve Approximation of Outer Ear

Concentric Circles for inner Ear

Proposed Approaches (Fusion techniques)

- Feature Fusion

  - Fuzzifier to assign a different confidence level to every input feature vector
  - Ranking all of the useful data for feature selection.
  - Feature reduction techniques to extract a small number of from the larger set of features

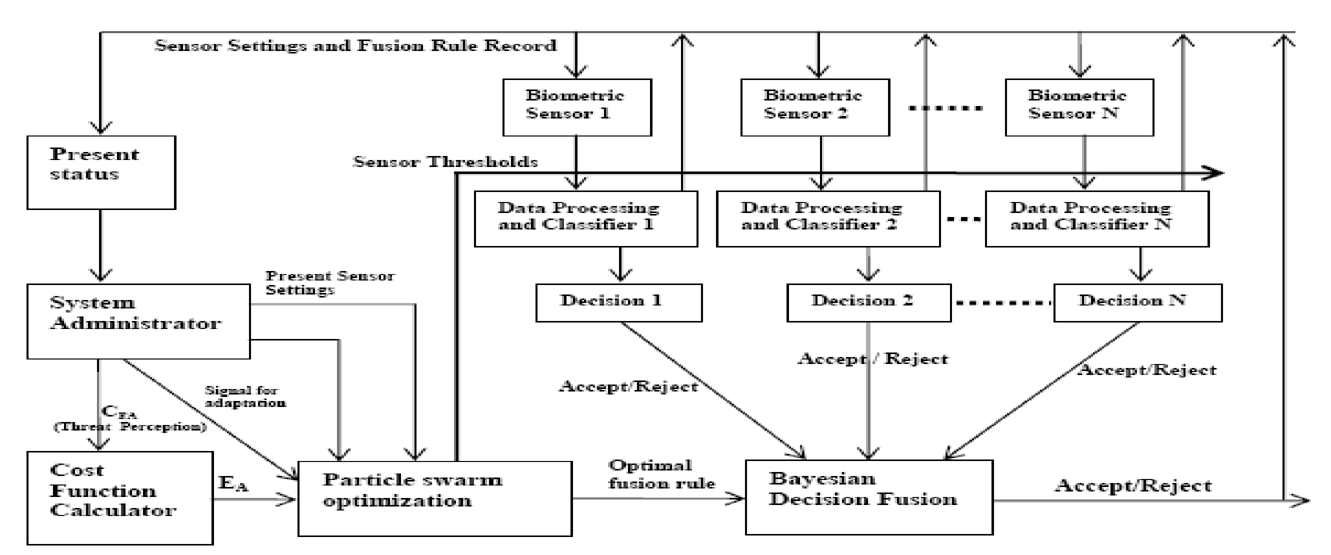


Final Features

Feature Level Fusion

Proposed Approaches (Fusion techniques)

- Decision Fusion
  - Error rates based Decisions (FAR/FRR)
  - Bayesian cost (a weighted sum of FAR and FRR) based objective function and Binary Fusion Rules based integration
  - Evolutionary swarm intelligence algorithm (PSO) based Parameters selection



Decision Level Fusion

THANKS

THANKS