Particle Swarm Optimization for Fuzzy Models

Arun Khosla

Department of Electronics & Communication Engineering
National Institute of Technology, Jalandhar - 144 011. INDIA
khoslaak@nitj.ac.in

1

Acknowledgements

- **Prof. (Dr.) K.K. Aggarwal**
  Vice-Chancellor
  GGS Indraprastha University, Delhi, India
- **Prof. (Dr.) Shakti Kumar**
  Society for Education and Research, Jagadhari, India
- **Jagatpreet Singh**
  Qualcomm, USA
- **Kumar Rahul Ghosh**
  Aricent, Gurgaon, India

2

Agenda

- Why Fuzzy Logic?
- Important Terms and Definitions
- Working of a Fuzzy Inference System
- Fuzzy Models Identification Problem
- PSO Preliminaries
- Identification of Fuzzy Models through PSO
- Taguchi Method
- Case Study

3

Prof. Lotfi A Zadeh

Department of Electrical &
Computer Engineering
University of California, Berkeley

Father of Fuzzy Logic

4

SOME APPLICATION AREAS

- **CONSUMER PRODUCTS (CAMERAS, CAMCORDERS, WASHING MACHINES etc.)**
- **INDUSTRIAL PROCESS CONTROL**
- **PATTEREN RECOGNITION**
- **MEDICINE**
- **MANAGEMENT**
- **FINANCE**

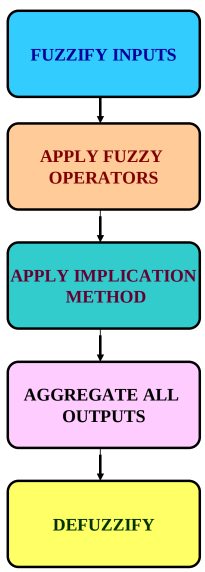
5

Advantages of Fuzzy Logic

- Lower cost of development
- Close to human intuition
- Conceptually easy to understand
- Flexibility

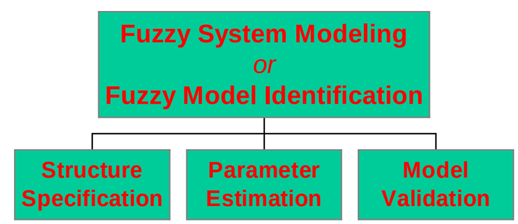
6

Working of Fuzzy Inference System



7

Fuzzy Modeling Process



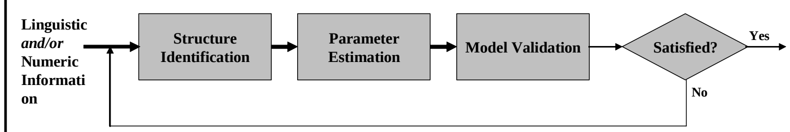
8

Fuzzy Modeling Identification: Some Issues

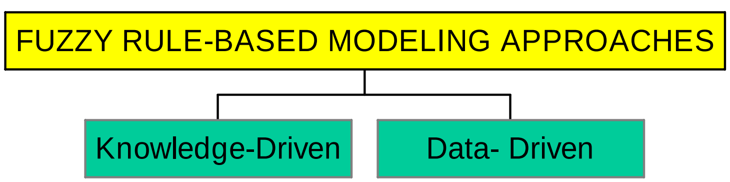
- Selecting the type of fuzzy model
- Selecting input and output variables for the model
- Choosing the structure of membership functions
- Determining the number of fuzzy rules
- Identifying the parameters of antecedent and consequent membership functions
- Defining some performance criteria for evaluating fuzzy models

9

Fuzzy Modeling Identification Process



10



11

BIOLOGICALLY INSPIRED FUZZY MODELING TECHNIQUES

- Genetic Algorithms
- Neural Networks
- Immune Algorithms
- Swarm Intelligence
  - Particle Swarm Optimization
  - Ant Colony Optimization

12

Swarm Systems

Swarm Systems

Swarm Systems

Swarm Intelligence

- A property of a system of unintelligent (seemingly dumb) agents of limited individual capabilities, exhibiting intelligent behavior.
- Intelligent behavior emerges generally from indirect communication and collaboration

Particle Swarm Optimization: Preliminaries

17

Inspiration for the PSO

BIRDS
to
BOIDS

18

Boids

- Movements of flocks of birds
  - simple in concept, yet visually complex
  - seem random, yet synchronized
- Flocking of birds investigated by Craig Reynolds.
- A formula derived to describe the movement of birds.
- Simulation of virtual birds, called **boids**, based on this formula.

19

Boid Algorithm

- A **clumping** force keeps the flock together.
  - i.e. each boid wants to be in the center of the flock.
- Each boid tries to **match velocity** with the rest of the flock.
- The boids try to **avoid collisions** with other boids.

20

Origin of PSO

- Kennedy and Ebenhart devised a way to use the Boids concepts for optimization.
- After modifying the Boids algorithm to eliminate unnecessary details, the Particle Swarm Algorithm was obtained.

PSO

- PSO can be easily implemented, and is computationally inexpensive**.**
- Population of potential solutions is used to probe the search space.
- **No** crossover or mutation.
- Particles change their position in search space instead of generating new particles.

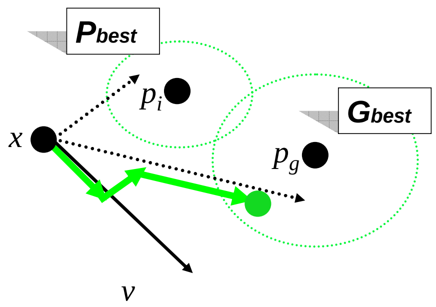
Properties of Particles

- Each particle is situated at a point in the search space. i.e. it has a position vector.
- An adaptive velocity step determines the next position at each iteration.
- Particles have memory, which stores
  - Co-ordinates of the Best Position the particle has visited
  - Co-ordinates of the Best particle in the flock.

The PSO Algorithm

- The movement of each particle is governed by
  - Its present Position and Velocity (momentum part).
  - Acceleration towards the previous best position (cognitive part).
  - Acceleration towards the best particle in the swarm (social part).

The PSO Algorithm



25

Weighted PSO

Weight factor

Two acceleration constants

$$
\begin{aligned}
v_{id}^{n+1} &= \chi\left(w v_{id}^{n} + c_1 r_1^{n}(p_{id}^{n} - x_{id}^{n}) + c_2 r_2^{n}(p_{gd}^{n} - x_{id}^{n})\right),\\
x_{id}^{n+1} &= x_{id}^{n} + v_{id}^{n+1},
\end{aligned}
$$

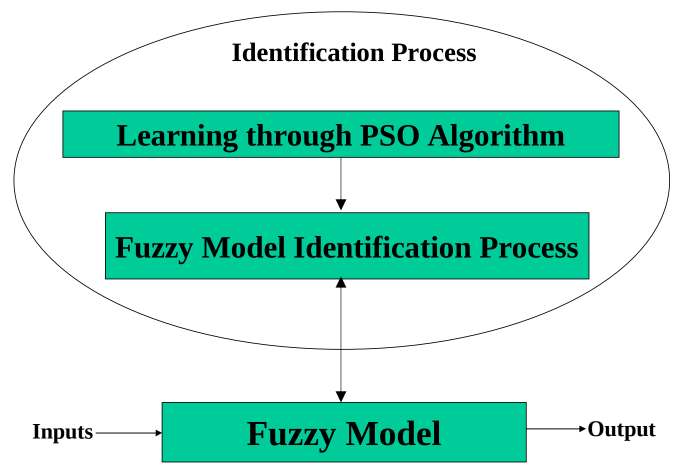
Constriction Factor

26

Many variants of PSO Algorithm has been proposed

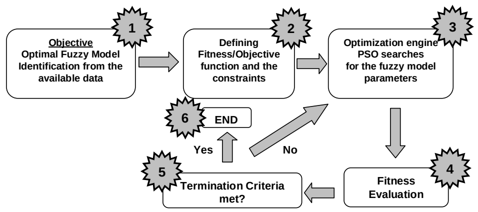
27

PSO for Fuzzy Models Identification



28

Fuzzy Model Identification using PSO as an Optimization Engine



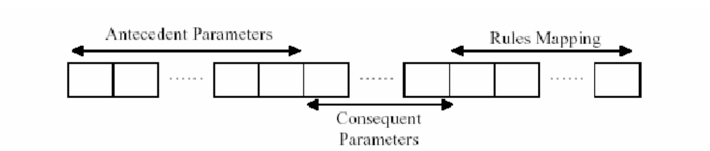
29

Types of Fuzzy Models

- Mamdani-type fuzzy models
- Takagi-Sugeno fuzzy models
- Singleton fuzzy models

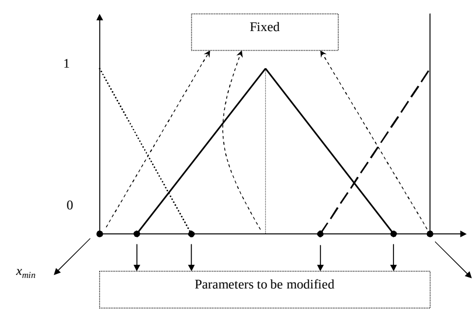
30

Encoding Mechanism



31

Encoding Mechanism



32

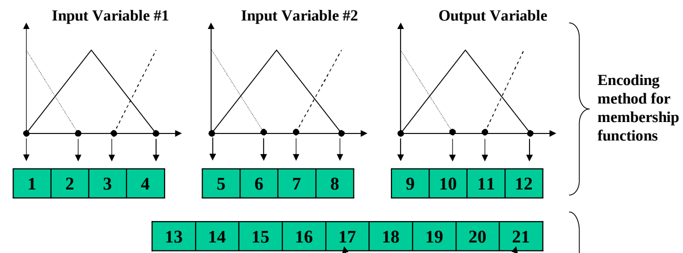
												Particle Dimensions
Input Variable #1	x^l_{11}	x^c_{11}	x^r_{11}	x^l_{12}	x^c_{12}	x^r_{12}	…	…	$x^l_{1m_1}$	$x^c_{1m_1}$	$x^r_{1m_1}$	$2m_1$-2
Input Variable #2	x^l_{21}	x^c_{21}	x^r_{21}	x^l_{22}	x^c_{22}	x^r_{22}	…	…	$x^l_{2m_2}$	$x^c_{2m_2}$	$x^r_{2m_2}$	$2m_2$-2
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
Input Variable #n	x^l_{n1}	x^c_{n1}	x^r_{n1}	x^l_{n2}	x^c_{n2}	x^r_{n2}	…	…	$x^l_{nm_n}$	$x^c_{nm_n}$	$x^r_{nm_n}$	$2m_n$-2
Output Variable	y^l_1	y^c_1	y^r_1	y^l_2	y^c_2	y^r_2	…	…	y^l_t	y^c_t	y^r_t	$2t$-2
Rule base (rule number)	R(1)	R(2)	R(3)	R(4)	R(5)	R(6)	…	…	R(m_1*m_2*…*m_n)			$\prod_{i=1}^{n} m_i$

Particle dimensions for representing Mamdani fuzzy model $= \sum_{i=1}^{n}(2m_i - 2) + (2t-2) + \prod_{i=1}^{n} m_i$

The shaded blocks represent the parameters modified through PSO

33

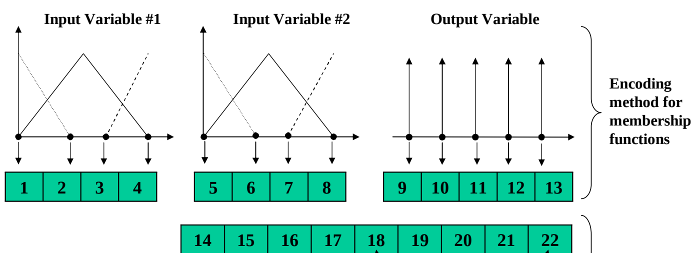
Particle Representing Mamdani Fuzzy Model



Ct		dT/dt		
		1	2	3
T	1	Rule 1	Rule 2	Rule 3
	2	Rule 4	Rule 5	Rule 6
	3	Rule 7	Rule 8	Rule 9

34

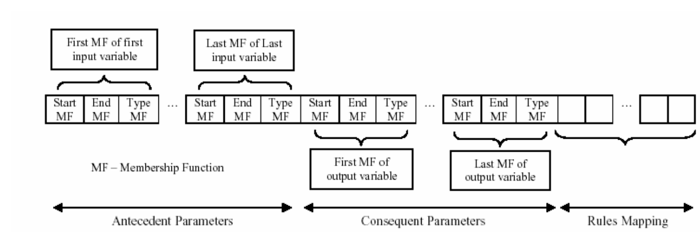
Particle Representing Singleton Fuzzy Model



Ct		dT/dt		
		1	2	3
T	1	Rule 1	Rule 2	Rule 3
	2	Rule 4	Rule 5	Rule 6
	3	Rule 7	Rule 8	Rule 9

35

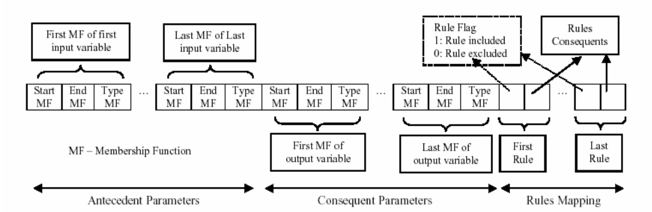
Particle Representing Mamdani Fuzzy Model (Case II)



Particle representing Mamdani fuzzy model where MF parameters, MF types and rule consequents can be modified through PSO Algorithm

36

Particle Representing Mamdani Fuzzy Model (Case III)



Particle representing Mamdani fuzzy model where MF parameters, MF types, rule consequents and rule-set can be modified through PSO Algorithm

A Framework for Fuzzy Models Identification through PSO

```
{
Define operating/strategy parameters for PSO algorithm;
Iteration  = 0;
Create initial swarm of particles;

     while iteration ≤ Number of iterations (or some other termination criteria)
          {
               Constrain Swarm;
               Build fuzzy model for each particle;
               Evaluate each fuzzy model and calculate MSE using (3.1)
               Update all the particles as per (5.1) and (5.2)
               Iteration=Iteration+1;
          }
     end
}
```

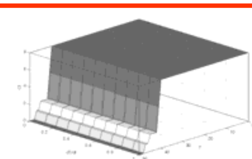

Simulation Results

Swarm Size	30
Iterations	2500
c_1	2
c_2	2
w_{start} (Inertia weight at the start of PSO run)	0.9
w_{end} (Inertia weight at the end of PSO run)	0.3
Vmax	75

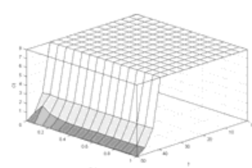
SIMULATION RESULTS

Model	MSE of Fuzzy Model corresponding to Swarm's *gbest*		Simulation time
	After 1st Iteration	After 2500 Iterations	
Mamdani	12.10	0.0488	19.424 hours
Singleton	46.95	0.1118	16.633 hours

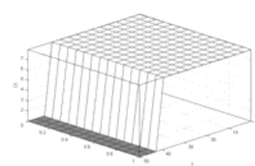
Surface Plots



Surface plot generated from the input-output data

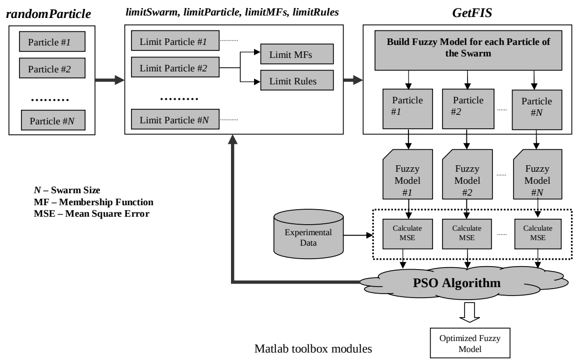


Surface plot for the identified Mamdani fuzzy model



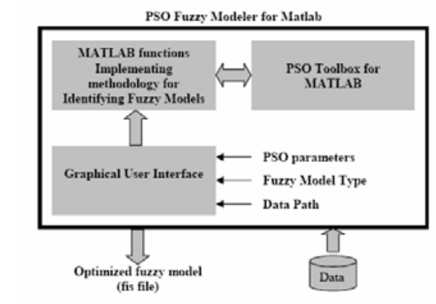
Surface plot for the identified Singleton fuzzy model

PSO Fuzzy Modeler for Matlab (Toolbox)



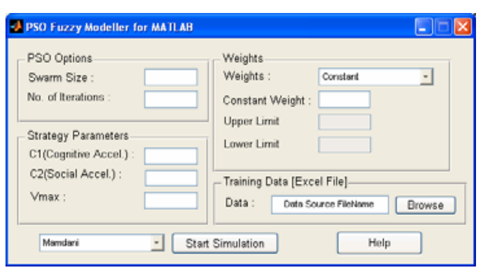
41

PSO Fuzzy Modeler for Matlab (Organization of various modules)



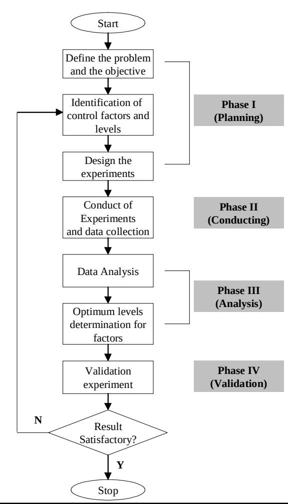
42

PSO Fuzzy Modeler for Matlab (GUI)



43

Taguchi Method for Identification of PSO parameters for improvement of Fuzzy Models



44

Some Features of Taguchi Method

- The fundamental principle of Taguchi method, which is an important tool for robust design, is to improve the quality of a product by minimizing the effect of the causes of variation without eliminating the causes
- Two major tools used in the Taguchi method are the orthogonal array (OA) and the signal to noise ratio (SN Ratio)
- OA is a matrix of numbers arranged in rows and columns. Each row represents the level of factors in each run and each column represents a specific level that can be changed for each run

45

Some Features of Taguchi Method (Contd.)

- The array is called orthogonal because all columns can be evaluated independently of one another
- SN Ratio is indicative of quality and the purpose of the Taguchi experiment is to find out the best level for each operating parameter such that SN Ratio is maximized (or minimized)
- The OAs of Taguchi method are fractional factorial designs that are used to study a large number of parameters with a small number of experiments. On the other hand, the full factorial design which represents the traditional or classical approach requires running all possible combinations

46

Taguchi Method

$$L_a(b^c)$$

Number of experiments

Number of levels

Number of columns

Orthogonal Array

47

Factors, Corresponding Parameters and their levels

Factor	Corresponding Strategy Parameter of PSO
A	c_1
B	c_2
C	w_{start}
D	w_{end}
E	V_{max}

Factor	Level			
	1	2	3	4
A	0.5	1	1.5	2
B	0.5	1	1.5	2
C	0.9	1	1.5	2
D	0.1	0.2	0.3	0.4
E	50	75	100	125

48

Matrix Experiments

Experiment Number	Factor						
	A	B	C	D	E	MSE	SN Ratio (10/MSE)
1	1	1	1	1	1	10.229	0.9776
2	1	2	2	2	2	10.2068	0.9797
3	1	3	3	3	3	0.3122	32.0307
4	1	4	4	4	4	8.6988	1.1496
5	2	1	2	3	4	9.8942	1.0107
6	2	2	1	4	3	6.8983	1.4496
7	2	3	4	1	2	0.1479	67.6133
8	2	4	3	2	1	3.584	2.7902
9	3	1	3	4	2	0.1105	90.4977
10	3	2	4	3	1	0.3739	26.7451
11	3	3	1	2	4	7.1799	1.3928
12	3	4	2	1	3	0.0389	257.0694
13	4	1	4	2	3	0.3359	29.7708
14	4	2	3	1	4	0.3094	32.3206
15	4	3	2	4	1	6.4665	1.5464
16	4	4	1	3	2	0.0489	204.4990

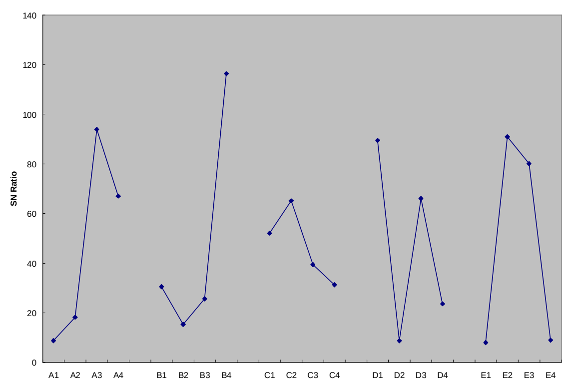
49

Response Table

Level	Factor				
	A	B	C	D	E
1	8.7844	30.5642	52.0798	*89.4952*	8.0148
2	18.2159	15.3737	*65.1515*	8.7334	*90.8974*
3	*93.9263*	25.6458	39.4098	66.0714	80.0801
4	67.0342	*116.3770*	31.3197	23.6609	8.9684

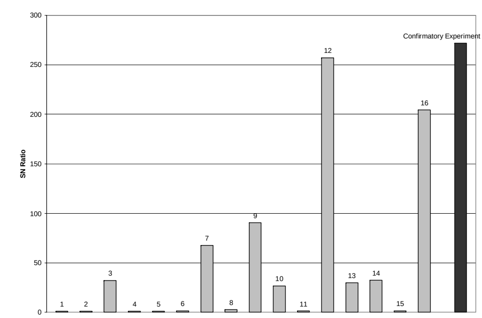
50

Response Graphs



51

Taguchi Method for optimization of PSO parameters for improvement of Fuzzy Models



52

Comparison of Computational Efforts

	Full Factorial Design (Traditional)	Fractional Factorial Design (Taguchi Method)
Time for 1 experiment	19.424 hours	19.424 hours
Total number of experiments (5 factors, each with 4 levels)	1024 (4^5)	16 (with $L_{16}(4^5)$ OA)
Total time for experimentation	828.16 days	12.94 days

References

- John Yen and Reza Langari, Fuzzy Logic Intelligence, Control, and Information, Prentice Hall, 1999
- J. Kennedy and R. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.
- Riza C. Berkan and Sheldon L. Trubatch, Fuzzy Systems Design Principles, IEEE Press, 1997
- K.E. Parsopoulos and M.N. Vrahatis, “Recent approaches to Global Optimization Problems through Particle Swarm Optimization”, Natural Computing, Kluwer Academic Publishers, pp.235-306, 2002.
- Eberhart, R.C and Shi, Y.,Particle, “Swarm Optimization: Developments, Applications and Resources”, Proceedings of the Congress on Evolutionary Computation, Seoul, Korea. pp. 81-86, 2001.
- Y.Shi, R. Eberhart and Y.Chen, Implementation of Evolutionary Fuzzy Systems, IEEE Transactions on Fuzzy Systems, pp. 109-119, April 1999.
- Genichi Taguchi, Subir Chowdhury and Yuin Wu, Taguchi Quality Engineering Handbook, John Wiley and Sons, 2005.

THANK YOU