Int. J. Open Problems Compt. Math., Vol. 3, No. 1, March 2010
ISSN 1998-6262; Copyright © ICSRS Publication, 2010
www.i-csrs.org

Fuzzy Membership Function Generation using Particle Swarm Optimization

Kurniawan Eka Permana and Siti Zaiton Mohd Hashim

Faculty of Engineering, Trunojoyo University, Madura, Indonesia
e-mail: kurniawan@if.trunojoyo.ac.id
Faculty of Computer and Information System, University Technology Malaysia, 81310, UTM Skudai, Johor, Malaysia
e-mail: sitizaiton@utm.my

Abstract

In this paper, we will propose a method to generate fuzzy membership function automatically. Particle Swarm Optimization is used as optimized algorithm, supplement the performance of fuzzy system. PSO is able to generate an optimal of fuzzy set for the membership functions automatic adjustment. Fuzzy control system that automatically back up a truck to a specified point on a loading dock is used as case study to validate the method. The objective of the problem is to back up the truck in order to go at specified position from any start position. Fuzzy system performance after generation show better result than before generation.

Keywords: *Fuzzy Control, Fuzzy Membership Function, Fuzzy Generation, Particle Swarm Optimization, Truck Backer Upper Problem.*

1 Introduction

Fuzzy systems are used in a wide number of applications areas especially on fuzzy control problems. One of standard control problem is the truck backer-upper problem. The object of the problem is to back up the truck so that it arrive perpendicular to the dock at specified position from any start position. These systems can be considered as knowledge-based systems, incorporating human knowledge into their knowledge base through fuzzy inference system and fuzzy membership functions. The definition of these fuzzy inference system and fuzzy membership functions is generally affected by subjective decisions, having a great influence over the performance of the system.[Esmin AA, 2007]

In recent years, some methods have been presented to generate fuzzy rules and membership functions. [Chiang and Lin, 1994] presented a method for applying the fuzzy set theory to teaching assessment. They apply the fuzzy set theory to propose a fuzzy statistical method to improve the classical statistics of the teaching assessment. [Echauz and Vachtsevanos, 1995] presented a fuzzy grading system. Their method can detect fairness by utilized students' and instructor's performance measures. [Juang and Wang, 2008] introduced the combination of Ant and PSO to generate fuzzy rules.

The PSO optimization technique was introduced by [Kennedy and Eberhart, 1995] as a stochastic search through an *n*-dimensional problem space aiming the minimization (or maximization) of the objective function of the problem. The PSO was built through the attempt to graphically simulate the choreography of a flock of birds flying to resources. Later, looking for theoretical foundations, studies were realized concerning the way individuals in groups interact, exchanging information and reviewing personal concepts improving their adaptation to the environment. PSO yields faster convergence when compared to Genetic Algorithm, because of the balance between exploration and exploitation in the search space [Visalakshi and Sivanandam, 2009].

[Esmin AA, 2007] has shown an efficient PSO based approach to construct a fuzzy rules base from data example. This method believe that the fuzzy logics can be also formulated as a space problem, where each point of fuzzy sets corresponds to a fuzzy logic i.e. represent membership functions, rule base and hence the corresponding system behavior.

In this paper, we present a method for constructing membership functions. The proposed method adjusts membership function automatically based on Particle Swarm Optimization. The parameter values to be optimized are the center, left and right of each foot membership function. After particles achieve the optimal result, the parameter value will be optimized by PSO and will be used to build the whole new fuzzy membership function

This paper is organized as follows: Section II presents an overview about the Particle Swarm Optimization (PSO). Section III presents the Fuzzy PSO generation. Section IV shows the problem to be solved. Section V shows the tests performed with a fuzzy control, which membership functions were adjusted using PSO algorithms. Finally, Section VI presents the conclusion.

2 Particle Swarm Optimization

The Particle Swarm Optimization Algorithm (PSO) is a population-based optimization method that finds the optimal solution using a population of particles [Kennedy and Eberhart, 1995]. Every swarm of PSO is a solution in the solution space. PSO is basically developed through simulation of bird flocking. The PSO definition is presented as follows:

Each individual particle *i* has the following properties: A current position in search space, x_{id}, a current velocity, p_{id}, and a personal best position in search space, pid.

• The personal best position, p_{id}, corresponds to the position in search space where particle *i* presents the smallest error as determined by the objective function *f*, assuming a minimization task.

• The global best position denoted by represents the position yielding the lowest error amongst all the p_{gd}.

During the iteration every particle in the swarm is updated using equations

Theorem 2.1 *Particles Velocity*

$$v_{id}=w*v_{id}+c1*rand()*(p_{id}-x_{id})+c2*rand()*(p_{gd}-x_{id})$$

The current position of the particle is updated to obtain its next position:

Theorem 2.2 *Particles Position*

$$x(t+1) = x(t) + v(t+1)$$

where *c1* and *c2* are two positive constants, *c1* and *c2* are two random numbers within the range [0,l], and *w* is the inertia weight.

3 Fuzzy PSO Model Generation

All the needed information about the rule-base and membership functions is required to be specified. These are used in order to particle to completely represent a fuzzy logic. For the model formulation and the implementation, consider a multi-input single-output (MISO) system with n number of inputs. The number of fuzzy sets for the inputs are m1 , m2 ,….., mn respectively.

However, it is unlikely that another research were taking few positions, as they were only partially and not fully position. [*Esmin, 2007*] use two sections of fuzzy sets which are left and right value. For adequate complementation of these algorithms with other, all the position is used to optimize. The dimensions of the particle for representing the fuzzy model can be worked out from the figure 4.1, which represents the membership functions for any one of the input/output variables with three membership functions..

3.1 Model Formulation

Several assumptions are used for the model formulation. These assumptions must be define and available before as a basic integration of this hybrid algorithm. The assumptions are listed as below:

(i) Triangular membership functions were used for both input and output variables with their centres are placed over the universe of discourse.

(ii) First and last membership functions of each input and output variable were represented with left- and right-skewed triangles

(iii) Complete rule-base was considered. A rule-base is considered complete when all possible combinations of input membership functions of all the input variables participate in fuzzy rule-base formation.

The integration between optimization algorithms and fuzzy logic problem is as follow:

1. The parameters are the centers, lefts and rights of each foot fuzzy membership function.
2. These parameters act as particles and looking for the global best fitness.
3. It starts with initial set parameters
4. After the parameters had been adjusted using optimization method, this parameter will be used to check the performance of the fuzzy logic.
5. The process is repeated until goal is achieved or optimization method reached the global best.

Optimization method like fig. 3.1 starts with the initial set parameter and gets the fitness function to define new values, representing the membership function parameter set of values. This new values will be used by the problem cases.

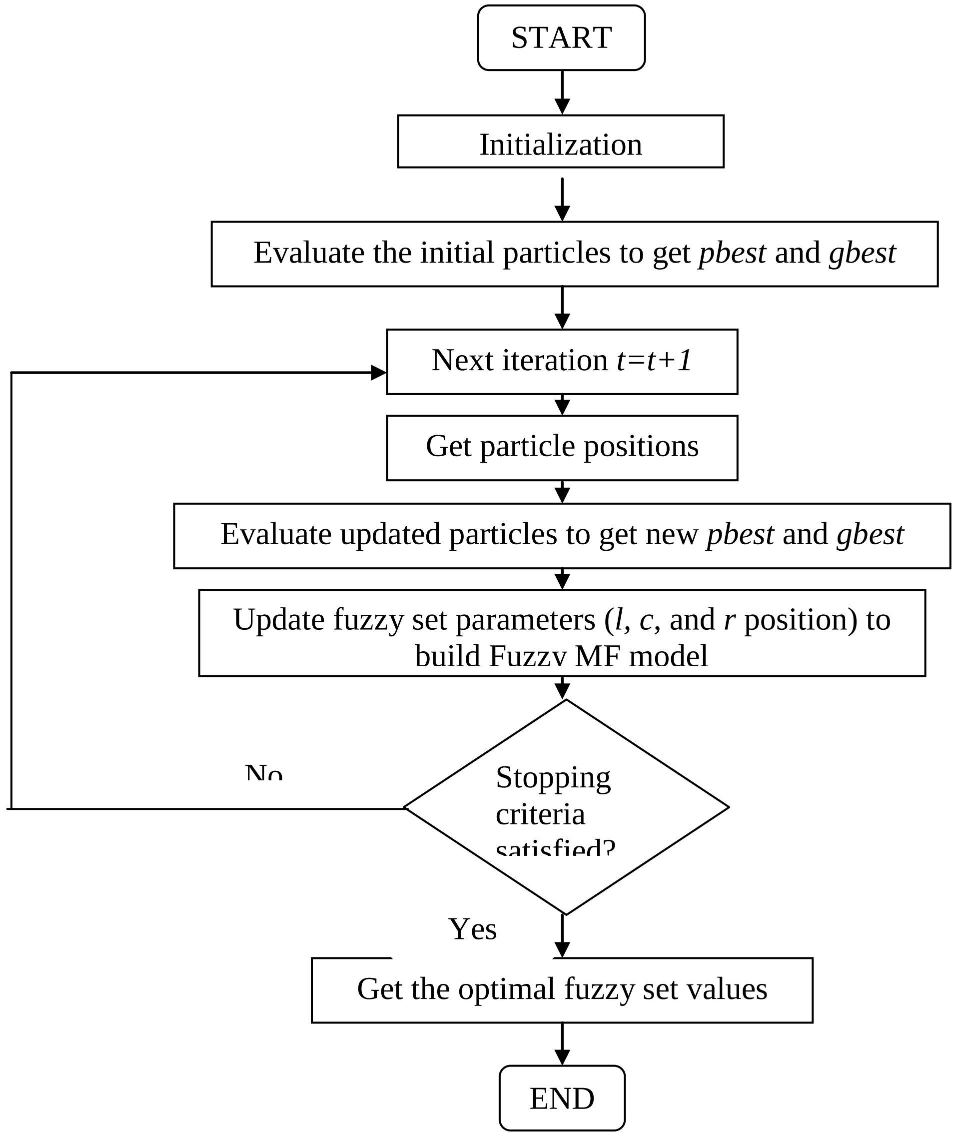

Fig. 3.1. Flowchart of Particle Swarm Optimization to adjust Fuzzy Membership Function

3.2 Particles Representation

Parameter value of membership function will be represented with particle in PSO. All input and output membership function changes based on the position update. The particle size for representing the membership functions of input variables for a model is given by theorem (3.1)).

Theorem 3.1 *Particles Dimension for Input Variables*

$$\sum_{i=1}^{n} (3m_i)$$

where, *n* - number of input variables and *m* - number of fuzzy sets

The particle size for representing the membership functions of output variables for a model is given by theorem (3.2).

Theorem 3.2 *Particles Dimension for Output Variable*

$$\sum_{i=1}^{n} (3t)$$

where, *n* - number of output variables and *t* - number of fuzzy sets.

The particle dimensions required for encoding the fuzzy model can be obtained in table 3.1

Table 3.1 Particle Dimension for Representing Fuzzy Model

	l	*c*	*r*	*l*	*c*	*r*	…	…	*l*	*c*	*r*	
Input variable #1	*x11*	*x11*	*x11*	*x12*	*x12*	*x12*	…	…	*x1m*	*x1m*	*x1m*	*3m1*
Input variable #2	*x21*	*x21*	*x21*	*x22*	*x22*	*x22*	…	…	*x2m*	*x2m*	*x2m*	*3m2*
…	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…
Input variable #n	*xn1*	*xn1*	*xn1*	*xn2*	*xn2*	*xn2*	…	…	*xnm*	*xnm*	*xnm*	*3mn*
Output variable	*y1*	*y1*	*y1*	*y2*	*y2*	*y2*			*yt*	*yt*	*yt*	*3t*

This particle dimension represents fuzzy membership function parameter value. First column show the input and output variable. In this column, number represents the input variable. Only one output variable is used here. The input variable is not limited to *n* variables. First row describe left, center and right of each membership function. The number of membership function represent until *m* variables. In the last column, 3m can be noted which means that 3 positions had been used and unlimited until *m* variables.

Fig. 3.2 shows triangle fuzzy membership function which has modified parameters. The membership function will shrinks, move or expand through the changes of each parameter value. The particle represent as the parameter value which keep changing until optimal value is reached.

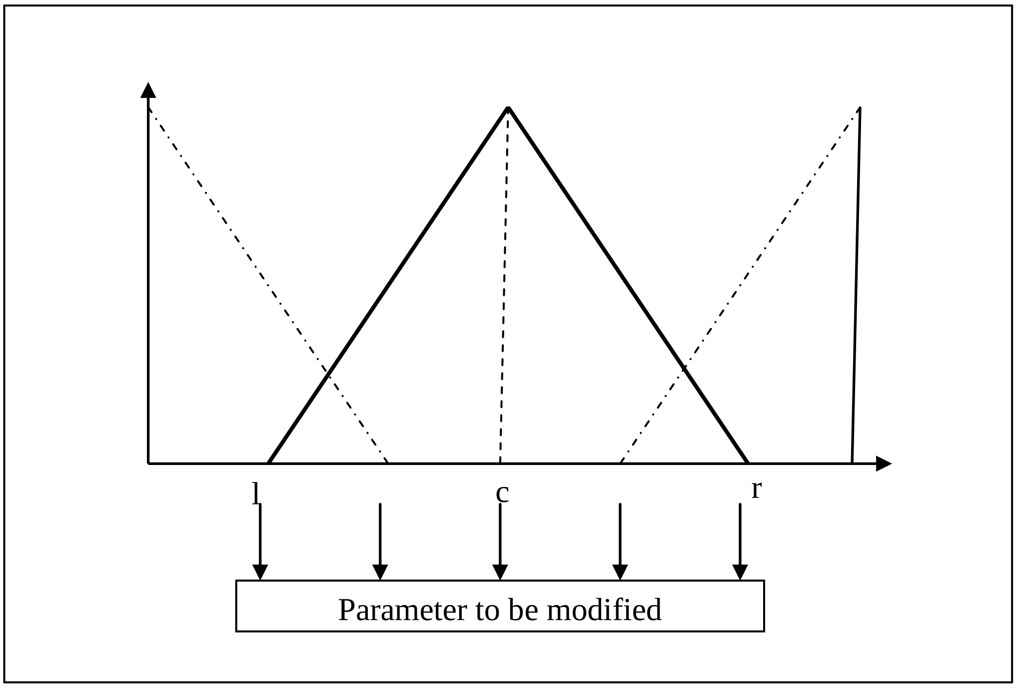

Fig. 3.2 Membership function parameter

After the optimal value is reached, the method will produce new parameter value and make different shape. When left parameter value is greater than before and right value is lesser than before but the center values fixed then shape will shrinks.

But when left parameter value is lesser than before and right value is greater than before then the shape will expand. Another shape will be moved when all the left, centre and right values are greater than before. It is also not impossible that different shape will conduct for different value.

4 Problem to be Solved

The objective of the truck backer-upper problem is to back up the truck so that it arrive perpendicular to the dock at specified position from any start position [Kong and Kosko, 1990; Kong and Kosko, 1992; Freeman, 1994].

First, we need to develop a set of fuzzy control rules and membership functions, which will define the truck path. Fig. 4.1 shows a simple model. The objective of the control system is to back up the truck so that it arrive perpendicular to the dock at position (x_f, y_f). The coordinate is (50,100). The point (x, y) is at the center of the rear of the truck, f is the angle of the truck axis to the horizontal, and q is the steering angle measured from the truck axis. The controller takes as input the position of the truck, specified by the pair (x, f), and outputs the steering angle q.

The coordinate x ranges from 0 to 100, f ranges from -100 to 280 and q ranges from -30 to 30 as shown in table 5.1, 5.2, and 5.3.

Next is defined the truck angle, f, and the truck x-position coordinate, x. At every stage the fuzzy should produce the steering angle, q, which backs up the truck to the dock.

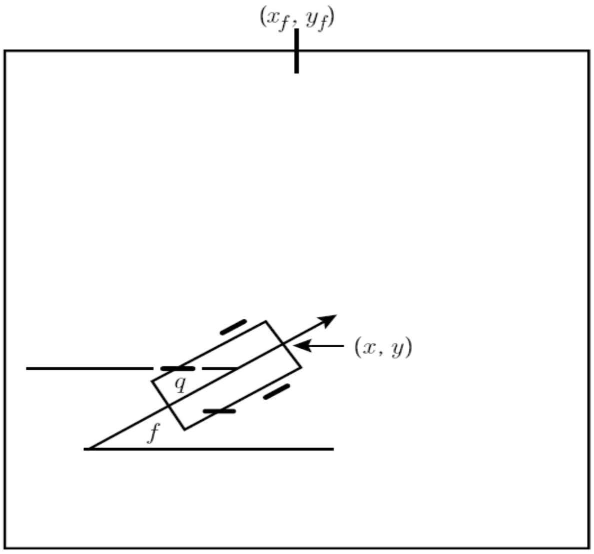

Fig. 4.1 The model for the simple truck backer upper [Freeman, 1994].

In truck backer upper problem, input x and f are represented in table 4.2 and table 4.3, while for output in table 4.4. The linguistic variables associated with the fuzzy sets for the x position are: LE (left), LC (left center), CE (center), RC (right center), and RI (right). Table 4.2 shows membership functions define those sets:

Table 4.2 Fuzzy Sets for *x* Position

MF	**Position**		
	left	***center***	***right***
LE	0	10	35
LC	30	40	50
CE	45	50	55
RC	50	60	70
RI	65	90	100

Fuzzy sets for the angle *f* are: RB (right below), RU (right upper), RV (right vertical), VE (vertical), LV (left vertical), LU (left upper), and LB (left below). The following membership functions define those sets as we can see in table 4.3.

Table 4.3 Fuzzy Sets for Angle

MF	**Position**		
	left	***center***	***right***
RB	-100	-45	10
RU	-10	35	60
RV	45	67,5	90
VE	80	90	100
LV	90	112,5	135
LU	120	155	190
LB	170	225	280

And then for the output variable steering angle, the sets are: NB (negative big), NM (negative medium), NS (negative small), ZE (zero), PS (positive small), PM (positive medium), and PB (positive big). Table 4.4 is shows fuzzy sets for steering angle.

Table 4.4 Fuzzy Sets for Steering Angle

MF	**Position**		
	left	***center***	***right***
NB	-30	-30	-18
NM	-25	-15	-5
NS	-12	-6	0
ZE	-5	0	5
PS	0	6	12
PM	5	15	25
PB	18	30	30

Finally, the fuzzy rules for this problem appear in Table 4.5. It is used to determine the output given by the input conditions in the form of if-then statement. Row 1 in table 4.5 represents the input x position and column 1 represent the angle *f*.

The fuzzy rules composed of 35 rules were determined by the input *x* position and the angle *f* [Freeman, 1994].

Table 4.5 Fuzzy Rules

	LE	**LC**	**CE**	**RC**	**RI**
RB	PS	PM	PM	PB	PB
RU	NS	PS	PM	PB	PB
RV	NM	NS	PS	PM	PB
VE	NM	NM	ZE	PM	PM
LV	NB	NM	NS	PS	PM
LU	NB	NB	NM	NS	PS
LB	NB	NB	NM	NM	NS

5 Result

Table 5.1 shows the parameter set for Truck Backer Upper Problem. This parameter has done in trial basis. For initial k_i and w_i , we use -1 and 5, respectively. Number of iteration is 500.

Table 5.1 PSO Parameter for Truck Backer Upper Problem

Parameter	**Value**
c1	1.45
c2	1.45
dt	0.1
Inertia weight	0.7
Number of particles	5

In this experiment, two different positions and angles have been tested and evaluated. First, the truck center is in the position *x*=50, *y*=20 and the angle *f*=30. The movement goes up and then reaches *x*-position 50 as a goal. Second, initial position is set to *x*=90, *y*=10 and the angle *f*=110. The movement also reaches *x*-position 50 as a goal.

	position x	*position y*	*angle f*
Case 1	50	20	30
Case 2	90	10	110
Destination	x=50	y=100	

Fig. 5.1 Position and angle

The x position is observed before and after experiment. It is found that tuning the fuzzy membership function has change the performance. Based on table 5.2 and 5.3, it is noted that each membership function changes from before.

It was shown that after optimization with PSO, our fuzzy membership function have different shapes. It looks like optimization with PSO make wider range according to the value.

Table 5.2 Fuzzy Sets For *X* Position After Optimization with PSO

	Position		
MF	***left***	***center***	***right***
LE	1.6	11.8	37.8
LC	30.9	41.9	52.9
CE	45.9	51.9	57.9
RC	50.9	61.9	72.9
RI	65.9	91.9	102.9

The experiment was repeated for next input, *y* position. It is noted that optimization on fuzzy membership function for *y* position was able to generate new membership function.

Table 5.3 Fuzzy Sets For *Y* Position After Optimization with PSO

	Position		
MF	***left***	***center***	***right***
RB	-100	-44	12.9
RU	-10	36.9	62.9
RV	45.9	69.4	92.9
VE	81	91.9	102.9
LV	91	114.4	137.9
LU	121	156.9	192.9
LB	171	226.9	282.9

Using the fuzzy membership function generated by optimization method (FPSO) and standard fuzzy, the performance of each was observed for truck backer upper problem. It is shown in fig. 5.2 and fig. 5.3, fuzzy control need more time and iteration to converge. FPSO has reached faster in term of iteration to converge.

From the result given, we can see for the case 1(50,20,30), Fuzzy and Fuzzy PSO have converged at iteration 41 and 32, respectively. FPSO also has achieved better precise value at position 51.56782 compare with 52.222 for fuzzy.

Second case (90, 10, 110) has achieved the same, FPSO achieve better convergence and closer to destination than fuzzy and get more precisely than standard fuzzy.

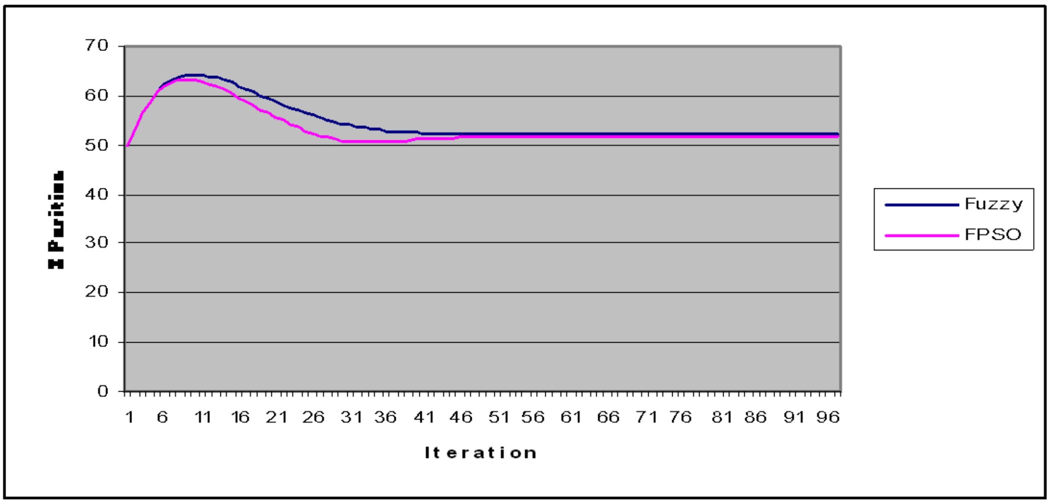

Figure 5.2 Comparison result for Case 1 between fuzzy and FPSO

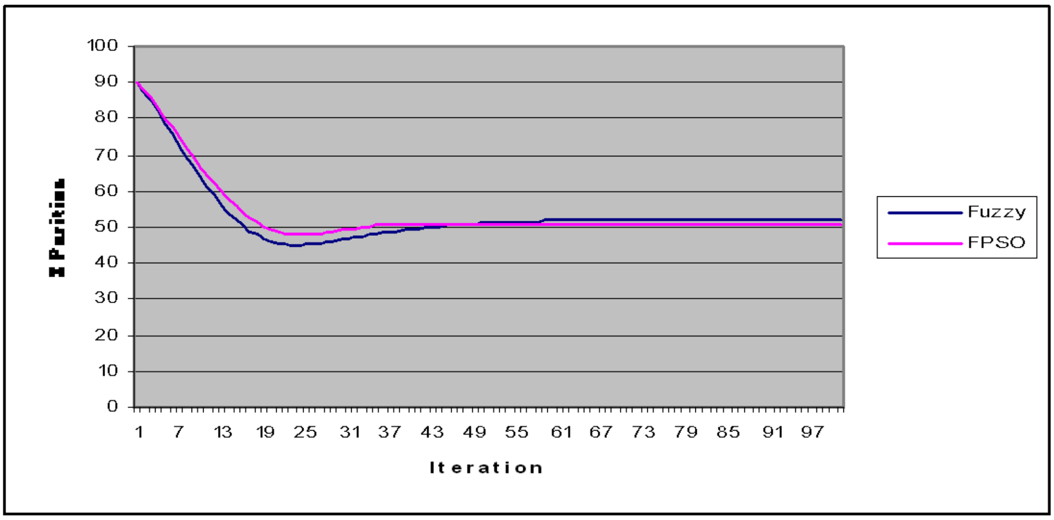

Figure 5.3 Comparison result for Case 2 between fuzzy and FPSO

6 Conclusions and Further Works

Fuzzy logic as the main basis in this research is the robustness of its interpolative reasoning mechanism and the deal with imprecise and incomplete information.

However, to set membership function in manual properly is time consuming, prone to errors and difficult, especially it depends so much to an expert.

The proposed method we use here called Fuzzy Particle Swarm Optimization (FPSO). It has successfully demonstrated its competence in generated membership function automatically. A whole new membership function successfully adjusted from standard fuzzy membership function. It could be done with representation of fuzzy membership function value as particles. In the each iteration in optimization method, the particle represent will be changes to reach the optimal value. The membership function will shrinks, move or expand through the changes of each value. Based on result experiment, the Fuzzy PSO has adjusted fuzzy membership function and improved the performance result in term accurately to destination and faster in speed of convergence.

The optimal result is presented as a great promise for optimization process. The authors intend to further investigate this problem in quest for development of an improved PSO, which can train the problem in faster time to converge

ACKNOWLEDGEMENTS.
The authors thank the Malaysian Government, Ministry of Higher Education (MOHE), Universiti Teknologi Malaysia for the Fundamental Research Grant Scheme (FRGS), Vot. No. 78257 and Soft Computing Research Group, FSKSM, UTM Malaysia.

References

[1] Eberhart, R. and Kennedy, I. “A new optimizer using particle swarm theory”.[J] Proc. 6 Int. Symposium on Micro Machine and Human Science, (1995).39-43.

[2] J. Kennedy and R.C. Eberhart, "Particle swarm optimization" , Proceeding of the 1995 IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Centre, Piscataway, NI, (1995), Iv: 1942-1948

[3] Y. Shi and R.C. Eberhart, "A modified Particle Swarm Optimiser", IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, (1998).

[4] Esmin, A. A. A, “Generating Fuzzy Rules from Examples Using the Particle Swarm Optimization Algorithm”, Hybrid Intelligent Systems, 2007. HIS (2007). 7th International Conference on 17-19 Sept. 2007 Page(s):340 – 343.

[5] S Michael Negnevitsky, Artificial Intelligence, Addison Wesley, (2004)

[6] S-G Kong and B. Koskos, “Adaptive Fuzzy Systems for Backing up a Truck-and-Trailer, IEEE Trans on Neural Networks, vol. 3, no.2, March, (1992)

[7] Freeman, James A., Fuzzy system for control applications : The Truck Backer Upper, The Mathematical Journal, (1994).

[8] Kong, Seong-Gon & Kosko, Bart, Adapative fuzzy system for backing up Truck and Trailler, IEEE Transactions on Neural Network vol.3 no.2 March (1992).

[9] Kong, Seong-Gon & Kosko, Bart, Comparison of Fuzzy and Neural Truck Backer Upper Control System, (1990)

[10] Hong, T. P., & Lee, C. Y. Induction of fuzzy rules and membership functions from training examples. Fuzzy Sets and Systems, 84(1), 33–47. (1996).

[11] Hong, T. P., & Lee, C. Y. Effect of merging order on performance of fuzzy induction. Intelligent Data Analysis, 3(2), 39–151. (1999).

[12] Ishibuchi, H., & Nakashima, T. Effect of rule weights in fuzzy rulebased classification systems. IEEE Transactions on Fuzzy Systems, 9(4), 506–515. (2001).

[13] Tsai, F. M., & Chen, S. M. A new method for constructing membership functions and generating fuzzy rules for fuzzy classification systems. In Proceedings of 2002 tenth national conference on fuzzy theory and its application, Hsinchu, Taiwan, Republic of China. (2002).

[14] Wu, T. P., & Chen, S. M. A new method for constructing membership functions and fuzzy rules from training examples. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics, (1999). 29(1), 25–40.

[15] P Visalakshi, S N Sivanadam, Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization, Int. J. Open Problems, Compt. Math, Vol 2, No 3, September, 2009

Permana KE received the Bachelor degree in Information Engineering from Sepuluh November Institute of Technology, Surabaya, Indonesia, in 2003. He is currently doing her Master degree in University Technology Malaysia, Johor, Malaysia. Currently she is a Lecturer in the department of Information Engineering, Trunojoyo University, Indonesia. His areas of interest are Machine Learning, Soft Computing and Fuzzy. He has published papers in National and International Conferences.

Dr.Siti Zaiton has obtained her PhD degree in Soft Computing in the year 2005 from University of Sheffield, UK. She was awarded Gold Award for her invention by Ministry of Science, Technology and Innovation Malaysia in the year 2006. Currently she is a lecturer in the Department of Software Engineering, University Technology Malaysia, Johor, Malaysia. She has published more than 20 research articles in International/National Journals. She has guided Master and PhD student.