Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

# Fuzzy Controller Design by Clustering-Aided Ant Colony Optimization

**Chia-Feng Juang*, Yu-Ping Kang** and Chiang Lo***

**Department of Electrical Engineering, National Chung Hsing University Taichung, Taiwan, R..O.C. ( e-mail:cfjuang@dragon.nchu.edu.tw).*
***Department of Electrical Engineering, Chung Chou Institute of Technology, Changhua, Taiwan, R.O.C.*

**Abstract:** This paper proposes a Clustering-aided Ant Colony Optimization (ACO) algorithm (CACO) for fuzzy controller design. The objective of CACO is to improve both the design efficiency of a fuzzy controllers and its performance. In CACO, the number of rules in CACO is created on-line by a newly proposed fuzzy clustering. Once a new rule is generated, the consequence is selected from a list of candidate control actions by ACO. In ACO, the tour of an ant is regarded as a combination of consequent actions selected from every rule. Searching for the best one among all consequence combinations involves using a proposed pheromone matrix and a new heuristic value assignment method. To verify the performance of CACO on fuzzy control, simulations on water bath temperature control are performed.

## 1. INTRODUCTION

Fuzzy control methodologies have emerged in recent years as promising ways to approach nonlinear control problems. However, a common bottleneck encountered in fuzzy controller design is that derivation of fuzzy rules is often difficult and time-consuming, and relies on expert knowledge. To overcome this disadvantage, many automatic methods for fuzzy controller design and optimization algorithms have been proposed (Wang, 1994, Cordon, *et al*, 2001).

The design of a fuzzy controller consists of designs of the antecedent part and the consequent part, where the former determines the number of fuzzy rules. Most studies that used genetic algorithms (GA) or particle swarm intelligence (PSO) for optimization of fuzzy controllers defined the number of fuzzy rules or the number of membership functions for each input variable in advance (Homaifar and McCormick, 1995, Belarbi and Titel, 2000, Juang *et al*, 2000, Juang 2004). For automatic rule determination, a new clustering algorithm with the abilities of on-line learning and automatic generation of number of rules is proposed in this paper. In the following control problem considered in this paper, the data are generated only when on-line learning is performed. Thus, on-line clustering instead of off-line clustering is proposed.

In fuzzy controller design, one challenging design task is the determination of the consequent part. In this paper, the Ant Colony Optimization (ACO) algorithm is employed to design the consequent part. The ACO (Dorigo, *et a1*, 1996, 1997) is a relatively new optimization computation technique. It is a multi-agent approach to solving difficult combinational optimization problems such as the traveling salesman problem (TSP) (Dorigo, *et a1,* 1996, 1997). ACO algorithm is inspired by real ant colonies in nature. Real ants can find shorter path from their nest to the food source by depositing chemical substance (pheromone). The ACO algorithm has been successfully applied to difficult combinatorial optimization problems (Dorigo, 2004). In (Cassillas, et al, 2005), ACO has been used for fuzzy system design. However, in that work, the antecedent part is partitioned in grid-type and the heuristic values in ACO are obtained off-line from the training data. In the proposed CACO, a new pheromone matrix together with an on-line heuristic value assignment is proposed. Effects of fuzzy clustering and on-line heuristic value assignment on the performance of CACO are studied in Section 5 using simulations.

The rest of this paper is organized as follows. Section 2 describes the fuzzy controllers to be deigned by CACO. In Section 3, the antecedent part learning by CACO is introduced. The basic concept of ACO and its application to learning the consequent part in CACO is presented in Section 4. In Section 5, the performance of CACO is compared with that of other design methods. Finally, conclusions are drawn in Section 6.

## 2. FUZZY CONTROLLER

The fuzzy system to be designed is described in this section. The $i$th rule, denoted as $R_i$, in the fuzzy system is represented in the following form:

$$R_i: \text{If } x_1(k) \text{ is } A_{i1} \text{ And } \ldots \text{ And } x_n(k) \text{ is } A_{in}$$

$$\text{Then } y(k) \text{ is } u_i(k) \tag{1}$$

where $x_1(k) \sim x_n(k)$ are input variables, $y(k)$ is the control output variable, $A_{ij}$ is a fuzzy set, and $u_i(t)$ is a recommend action and is a fuzzy singleton. A Gaussian membership function is used for fuzzy set $A$ and the membership function is

$$M(x) = \exp\{-\frac{(x-m)^2}{\sigma^2}\} \tag{2}$$

where $m$ and $\sigma$ represent the center and width of the fuzzy set $A$, respectively. In the inference engine, the fuzzy AND operation is implemented by the algebraic product in fuzzy theory. Thus, given an input data set $\vec{x} = (x_1, \ldots, x_n)$, the firing strength $\phi_i(\vec{x})$ of rule $i$ is calculated by

$$\phi_i(\vec{x}) = \prod_{j=1}^{n} M_{ij}(x_j) = \exp\left\{-\sum_{j=1}^{n}\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right\} \tag{3}$$

Supposing that there are $r$ rules in a fuzzy controller, then the output of the system calculated using weighted average defuzzification method is

$$y = \frac{\sum_{i=1}^{r} \phi_i(\vec{x}) u_i}{\sum_{i=1}^{r} \phi_i(\vec{x})} \tag{4}$$

## 3. ANTECEDENT PART LEARNING

This section introduces the fuzzy rules and the fuzzy sets in each input variable generated by the proposed fuzzy clustering. Partitioning the antecedent part manually is time-consuming, especially when there are many inputs in a fuzzy system, and many redundant fuzzy rules are generated. Automatic generation of fuzzy rules by fuzzy clustering helps ease the design effort and avoid the generation of unnecessary rules.

Geometrically, a rule corresponds to a fuzzy cluster in the input space (Juang and Lin, 1998). For each incoming input pattern $\vec{x}$, the firing strength can be regarded as the degree of the incoming input pattern that belongs to the corresponding cluster. According to this concept, the firing strength $\phi_i(\vec{x})$ in Eq. (3) is used as the criterion for deciding if a new fuzzy rule should be generated. There are no rules initially. For the first incoming data $\vec{x}(0)$, a new fuzzy rule is generated, with the center and width of Gaussian fuzzy set $A_{1j}$ assigned as

$$m_{1j} = x_j(0) \text{ and } \sigma_{1j} = \sigma_{fixed}, \text{ for } j = 1, \cdots, n \tag{5}$$

where $\sigma_{fixed}$ is a pre-specified value that determines the width of each fuzzy set. This paper sets $\sigma_{fixed}$ at 0.4. For the succeeding incoming data $\vec{x}(k)$, find

$$I = \arg \max_{1 \le i \le r(k)} \phi_i(\vec{x}(k)) \tag{6}$$

where $r(k)$ is the number of existing rules at time $k$. If $\phi_I \le \phi_{th}$, where $\phi_{th} \in (0,1)$ is a pre-specified threshold, then a new fuzzy rule is generated and $r(k+1) = r(k) + 1$.

The number of fuzzy sets in each input variables is also increased by one. The center and width of the new fuzzy set are shown below,

$$m_{r(k+1)j} = x_j,\ \sigma_{r(k+1)j} = \sigma_{fixed} \tag{7}$$

## 4. CONSEQUENCE LEARNING BY ACO

ACO is a meta-heuristic algorithm inspired by the behavior of real ants, and in particular how they forage for food. ACO can be applied to problems that can be described by a graph, where the solutions to the optimization problem can be expressed in terms of feasible paths on the graph. Among the feasible paths, ACO can be used to find the one with minimum cost. The whole ACO algorithm can be described by taking the traveling salesman problem (TSP) as an example. The TSP is to find a minimal length with each city visited once. We are given a set of $N$ cities, represented by nodes, and a set $E$ of edges with fully connecting nodes $N$. Let $d_{ij}$ be the length of the edge $(i, j) \in E$, that is the distance between cities $i$ and $j$, with $i, j \in N$. At each iteration $t$, an ant in city $i$ has to choose the next city $j$ to head for from among those cities that it has not yet visited. The probability of picking a certain city $j$ is calculated using the distance between cities $i$ and $j$, and the amount of pheromone on the edge between these two cities. The probability with which an ant $q$ chooses to go from city $i$ to city $j$ is

$$p_{ij}^q(t) = \begin{cases} \dfrac{[\tau_{ij}(t)][\eta_{ij}]^\beta}{\sum_{\ell \in N_i^q} [\tau_{i\ell}(t)][\eta_{i\ell}]^\beta} & \text{if } j \in N_i^q \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

where $\tau_{ij}(t)$ is the amount of pheromone trails on edge $(i, j)$ at iteration $t$, $\eta_{ij} = 1/d_{ij}$ is the heuristic value of moving from city $i$ to city $j$, $N_i^q$ is the set of neighbors of city $i$ for the $q$ th ant, and parameter $\beta$ controls the relative weight of pheromone trail and heuristic value. After all ants have completed their tours, the pheromone level is updated by

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{9}$$

where $0 < \rho \le 1$ is the pheromone trail evaporation rate. The update value $\Delta\tau_{ij}$ is related to the quality value $F$ and many updating rules for $\Delta\tau_{ij}$ have been studied (Dorigo, 2004).

Application of ACO to the design of consequence of each generated fuzzy rule is introduced as follows. First, all of the

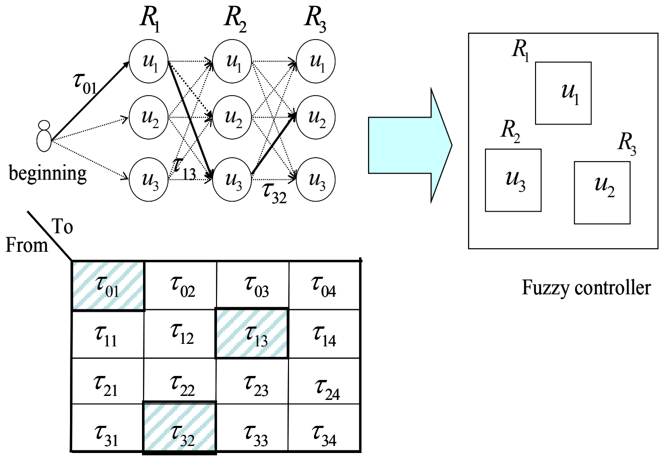

Fig. 1. The relationship between the tour of an ant, the selected consequent actions in a fuzzy system, and the pheromone matrix.

possible consequent actions are listed in a set $U=\{u_1,\cdots,u_N\}$. For each fuzzy rule, the number of candidate actions is $N$. Suppose there are $r$ rules, then the complexity of finding the best consequent combination is $N^r$. This is a combinational optimization problem, and the ACO algorithm is exploited to find the solution. The tour of an ant is regarded as one combination of consequent actions selected from every rule. For example, suppose there are three fuzzy rules generated by fuzzy clustering. For simplicity, assume that $N$=3. The relationship between ant's tour, selected consequence action, and corresponding pheromone matrix is shown in Fig. 1, where the three rules are denoted by $R_1$, $R_2$ and $R_3$, and the three candidate actions $u_1$, $u_2$, and, $u_3$ are denoted by nodes for each rule.

Starting from the initial state, the nest, the ant moves through $R_1$, $R_2$ and stops at R3, where the tour of this ant is marked by a bold line. For each rule, the node visited by the ant is selected as the consequent part of the rule. For the whole fuzzy controller constructed by the ant in Fig. 1, the consequent part values in $R_1$, $R_2$, and $R_3$ are $u_1$, $u_3$ and $u_2$, respectively. The tour from the current rule to the next rule is partially determined according to the pheromone trails between the rules, and the pheromone trails are stored in a pheromone matrix as shown in Fig. 1. Searching for the best one among all combinations of candidate actions involves partially the pheromone matrix. The size of the pheromone matrix is $N(N+1)$ and each entry is denoted by $\tau_{ij}$, where $i=0\sim N$ and $j=1\sim N$. The special value $\tau_{0j}$ represents the pheromone trail from an ant's initial state to the consequences of the first rule $R_1$. That is, it determines the selection of the control action of $R_1$. Then, for two neighboring rules $R_{m-1}$ and $R_m$, if the consequent action selected by $R_{m-1}$ is $u_i$, then the $N$ candidate control actions of $R_m$ is selected partially according to $\tau_{ij}$, $j=1\sim N$. This means that selection of the consequent part of each rule is dependent on the consequent part of other rules.

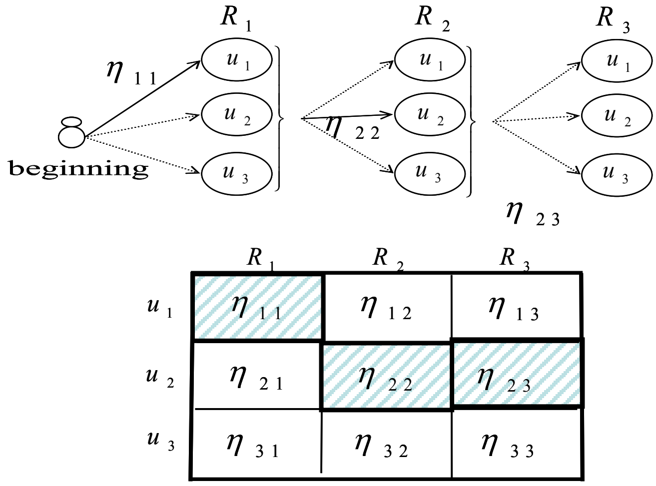

Fig. 2. The relationship between the tour of an ant and the heuristic value matrix, and notation of $\eta$ for each consequent action in each rule.

In addition to pheromone trail $\tau_{ij}$, the selection of the consequent action also depends on heuristic value $\eta$. In CACO, each candidate action of a fuzzy rule is assigned a heuristic value, i.e., there are $N$ heuristic values for each rule. As shown in Fig. 3, the heuristic value of candidate $u_j$ in rule $R_m$ is denoted by $\eta_{jm}$, where $j=1,\ldots,N$ and $m=1,\ldots,r$. In contrast to selection by pheromone matrix, the ant tour in Fig. 2 shows that consequence selection by heuristic information is independent of the actions selected by context fuzzy rules. The travel of an ant corresponds to the choice of the consequent actions of all rules, according to both the pheromone level and $\eta$. At iteration $t$, when the selected consequent action of current rule $R_{m-1}$ is $u_i$, then the probability of selecting candidate action $u_j$ of the following rule $R_m$ is given by

$$p_{ij}^m(t)=\frac{\tau_{ij}(t)(\eta_{jm}(t))^{\beta}}{\sum_{k=1}^{N}\tau_{ik}(t)(\eta_{km}(t))^{\beta}} \tag{10}$$

where $j=1,\ldots,N$. In a fuzzy controller, since it is natural for different rules to use the same control action, the set $N_i^q$ in Eq. (8) is released to the whole set $U$ for all $i$ in Eq. (10).

In the control problems considered in this paper, it is assumed that neither *a priori* knowledge of the plant model nor training data collected in advance are available. For this case, the heuristic values in CACO are updated on-line according to the difference between the actual output $y(k)$ and the desired output $y_d(k)$. For each fuzzy controller, the selected consequent actions are denoted as $u_{\hat{m}}$, $\hat{m} \in \{1, \dots, N\}$, for rule $R_m$. The selected controller is applied to the controlled plant. At each time step $k$, the control performance is measured by $e(k) = | y(k) - y_d(k) |$. The reward for $\eta$ update is defined as

$$\overline{R}(t,k) = \exp(-\varepsilon \cdot \frac{e(k)}{t}) \quad (11)$$

where parameter $\varepsilon$ controls the change rate of $\overline{R}(t,k)$. Equation (11) indicates that a larger reward is received for a smaller control error $e(k)$ and the reward is smaller for a smaller number of iterations $t$, thus avoiding fast convergence to poor solution. Since not only the current control performance, but the long-term control performance are also considered, the idea of temporal difference TD(0) for one-step Q-learning (Sutton, 1998) is used for $\eta$ update. From this temporal difference, $\eta_{\hat{m}m}$ is updated by

$$\eta_{\hat{m}m}(t,k+1) = \eta_{\hat{m}m}(t,k) + c_1 \cdot \frac{\phi_m(k)}{\sum_{i=1}^{r} \phi_i(k)} \cdot \left( \overline{R}(t,k) + \gamma \cdot \eta_{m^*m}(t,k) - \eta_{\hat{m}m}(t,k) \right)$$

$$, \; m = 1, \dots, r \quad (12)$$

where $\overline{R}(t,k) + \gamma \cdot \eta_{m^*m}(t,k) - \eta_{\hat{m}m}(t,k)$ is called the temporal difference error, $\gamma$ is the discount factor, $c$ is the learning parameter that controls the learning rate of $\eta$, and $\eta_{m^*m}(t,k) = \max_{1 \le j \le N} \eta_{jm}(t,k)$. The term $\phi_m / \sum_{i=1}^{r} \phi_i$ distributes temporal difference error to $\eta_{\hat{m}m}$, $m = 1, \dots, r$, according to the rule-firing strength. Parameters for $\eta$ update are set to be $c = 0.001$, $\varepsilon = 100$ and $\gamma = 0.2$. Parameter $\rho$ for pheromone trail update is set to be 0.1. These parameters are chosen heuristically and are appropriate different simulations examples we have performed.

The pheromone trails, $\tau_{ij}$, on the tour of the ant are updated according to the performance of the constructed fuzzy controller. When an ant completes a tour, the corresponding fuzzy controller is evaluated by a quality value $F$. A higher value of $F$ means better performance. For each iteration, after all the ants in the colony have completed their tours, i.e. the construction of $N_a$ fuzzy controllers, we find the one with the highest $F$ from the initial iteration till now. If a new global best ant is found in this iteration, then pheromone trails on the tour of the global best ant is updated; otherwise, no pheromone update is performed in this iteration.

Denote the global best ant as $q^*$ with the corresponding quality value as $F_{q^*}$. The new pheromone trail $\tau_{ij}(t)$ is updated by

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\textit{if } (i,j) \in \textit{global}\text{-best-tour} \quad (13)$$

where $0 < \rho < 1$ is the pheromone trail evaporation rate and

$$\Delta\tau_{ij}(t) = c_2 \cdot F_{q^*} \quad (14)$$

where $c_2$ is the parameter for controlling the amount of update.

## 5. SIMULATIONS

In this section, fuzzy systems design by CACO are simulated. Simulations examples include nonlinear plant tracking control, water bath temperature control and chaotic system control. In these examples, training data are generated only when control starts. In ACO, the number of ants is $N_a = 10$. Parameter $\beta$ in Eq. (10) is set to be 0.5. Parameter $\rho$ for pheromone trail update is set to be 0.1. These parameters are chosen heuristically so that a suitable performance is achieved.

*Example (Water bath temperature control)*

The plant of water bath temperature control (Tanomaru and Omatu, 1992) is described by

$$y(k+1) = a(T_s)y(k) + \frac{b(T_s)}{1 + e^{0.5y(k)-\gamma}}u(k) + [1 - a(T_s)]Y_0 \quad (15)$$

where

$$a(T_s) = e^{-\alpha T_s} \quad (16)$$

$$b(T_s) = \frac{\beta}{\alpha}(1 - e^{-\alpha T_s}) \quad (17)$$

and $u(k)$ is the control input of plant and is limited to 0 and 5V. The parameters in this example are $\alpha = 1.00151 * 10^{-4}$, $\beta = 8.67973 * 10^{-3}$, $\gamma = 40$, $Y_0 = 25(^\circ C)$, and the sampling period $T_s$ is 25 (sec.). The reference water temperature $y_{ref}(k)$ is set as follows,

$$y_{ref}(k) = \begin{cases} 35^\circ C & k \le 40 \\ 50^\circ C & 40 < k \le 90 \\ 65^\circ C & 90 < k \le 140 \end{cases} \quad (18)$$

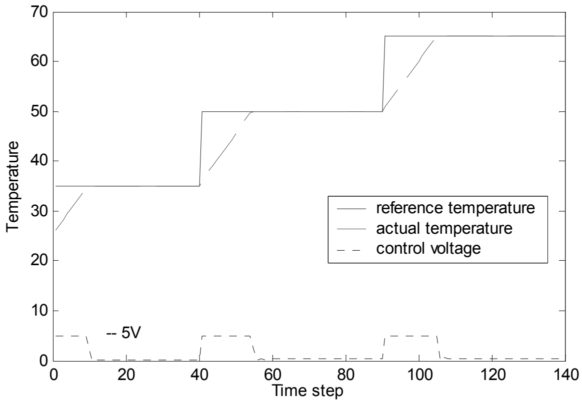


Fig. 3. Control results by CACO.

Table 1. Control Results of Fuzzy Controllers Designed by Different Methods in Example 2.

| | Manual (Lin, 1996) | SEFC (Juang, 2000) | NFIN (Lin, 1999) | CACO |
|---|---|---|---|---|
| Rule number | 25 | 7 | 7 | 7 |
| Control error | 369.5 | 325.3 | 296.6 | 287.3 |

Define $e(k) = y_{ref}(k) - y(k)$ and $\Delta e(k) = e(k) - e(k-1)$. The two input variables of the fuzzy controller are $x_1(k) = e(k)/40$ and $x_2(k) = \Delta e(k)/15$, so that they are in the range $[-1,\ 1]$. The set of candidate actions is $U = C_u \times \{0, \pm 0.1, \pm 0.2, \pm 0.3, \pm 0.4, \pm 0.8, \pm 1.2, \pm 1.6, \pm 2.0\}$ Since the actual control output is limited to 5V, when the fuzzy controller output is exceeds 5V, we should set it back to 5V. The control error and quality function are defined as

$$Error = \sum_{k=1}^{140} |e(k)|,\quad F = 1/Error \tag{19}$$

In CACO, parameters $\phi_{th}$ and $c_2$ are set to be 0.4 and 1, respectively. The influence of the range of candidate actions in $U$ is discussed. This examples simulates $C_u$ =10 and 100. The number of iterations is 1000 and there are 50 runs. After 50 runs, the average number of fuzzy rules generated is 9 and the average control errors are 344.1 and 287.2 when $C_u$=10 and 100, respectively. One control result with $C_u = 100$ is shown in Fig. 3, where the control error is 286.5, and there are seven rules.

Other fuzzy control methods that had been applied to the same temperature control problems are simulated and their control errors are shown in Table 1. These methods include fuzzy controller with manual design (See Appendix 2 of Lin, 1996), symbiotic evolution fuzzy control (SEFC) (Juang *et al* , 2000, and neural fuzzy inference network (NFIN) control (Lin *et al*, 1999). The results in Table 1 show that fuzzy controller designed by CACO achieves the minimum control error.

## 6. CONCLUSIONS

This paper proposes exploits the ant colony optimization algorithm and on-line fuzzy clustering for fuzzy controller design. In CACO, fuzzy rules are generated automatically and consequent actions are optimally selected. The use of fuzzy clustering eases the antecedent part design effort. For consequence design, simulation and comparison results in the examples have also verified the efficiency of using ACO, especially the effects of the proposed form of pheromone matrix and updating of heuristic values. In the future, more fuzzy control applications using CACO will be studied. .

## REFERENCES

Wang, L. X. (1994). *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice Hall.

Cordon, O., F. Herrera, F. Hoffmann and L. Magdalena (2001). Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, vol. 19 of *Advances in Fuzzy Systems – Applications and Theory*, World Scientific.

Homaifar, A. and E. McCormick (1995). "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol. 3, no. 2, pp. 129-139.

Belarbi, K.and F. Titel (2000). "Genetic algorithm for the design of a class of fuzzy controllers: an alternative approach," *IEEE Trans. Fuzzy Systems,* vol. 8, no. 4, pp. 398-405.

Juang, C. F., J. Y. Lin, and C. T. Lin (2000). "Genetic reinforcement learning through symbiotic evolution for fuzzy controller design, *IEEE Trans. Systems, Man, and Cyber.-Part B. Cybernetics*, vol. 30, no. 2, pp. 290-302.

Juang, C. F. (2004). "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst., Man, and Cyber., Part B: Cybernetics*, Vol. 34, No. 2, pp. 997-1006.

Dorigo, M., V. Maniezzo, and A. Colorni (1996). "Ant System: Optimization by a Colony of Cooperating Agents," *IEEE Trans. on Systems*, *man, and cybernetics, Part B: Cybernetics,* vol. 26, no. 1. pp. 29-41.

.

Dorigo, M. and L.M. Gambardella (1997). "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem" *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66.

Cassillas, J., O.Cordon, I. F. Viana (2005). "Learning cooperative linguistic rules using the best-worst ant system algorithm," *Int. Journal of Intelligent Systems*, vol. 20. pp. 433-452.

Dorigo, M, and T. Stü tzle (2004). Ant Colony Optimization, MIT.

Juang, C.F. and C.T. Lin, (1998). "An on-line self-constructing neural fuzzy inference network and its applications," *IEEE Trans. Fuzzy Systems,* vol.6, pp. 12-32.

Sutton, R. S. Sutton and A. G. Barto (1998). *Reinforcement Learning*, Chap. 6, The MIT Press.

Tanomaru, J. and S. Omatu (1992). "Process control by on-line trained neural controllers," *IEEE Trans. Ind. Electron*., vol. 39, pp. 511-521.

Lin, C. T. Lin, C.F. Juang, and C.P. Li (1999). "Temperature control with a neural fuzzy inference network," *IEEE Trans. Syst., Man, Cybern., Part C: Applications and Reviews,* Vol. 29, No. 3, pp. 440-451.

Lin, C. T. Lin and C. S. G. Lee (1996). *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*, Prentice Hall.

## Appendix. ACKNOWLEDGEMENT

This was supported by the National Science Council, Taiwan, Republic of China, under Grant number NSC 96-2628-E-005-087-MY3.