

Fuzzy Controller Design Using Group-Crossover Particle Swarm Optimization for Truck Reversing Control

Chia-Feng Juang, Yu-Cheng Chang, Chia-Hung Hsu
Department of Electrical Engineering
National Chung-Hsing University
Taichung, Taiwan, R.O.C
e-mail:cfjuang@dragon.nchu.edu.tw

I-Fang Chung
Institute of Biomedical Informatics
National Yang-Ming University
Taipei, Taiwan, R.O.C

Abstract—This paper proposes a fuzzy controller design using group-crossover particle swarm optimization (GCPSO) algorithm. The GCPSO uses a group-based framework for defining particle neighborhood topology and incorporating crossover operation into particle swarm optimization. The GCPSO dynamically forms different groups for selecting parents in crossover operations, particle updates and replacements. The objective of GCPSO is to improve fuzzy control accuracy. Comparisons with different population-based optimizations on truck reversing control problem demonstrate the performance of GCPSO algorithm.

Keywords- fuzzy controller, particle swarm optimization, genetic algorithms.

I. INTRODUCTION

Fuzzy controllers (FCs) have been successfully applied to many optimum control problems, like robots and power systems. Evolutionary fuzzy control (EFC) concerns automatic design of FC using evolutionary computation techniques. Two popular EFC approaches are genetic algorithms (GAs) [1,2] and particle swarm optimization (PSO) [3]. The PSO algorithm studies collective behavior in decentralized systems [4]. Its development was based on mimicking the social behavior of animals or insects models in an effort to find the optima in the problem space. Original PSO models are initialized with a population of random solutions. Each individual in PSO is assigned with a randomized velocity according to its own and its companions' flying experiences, and the individuals, called particles, are the flown through hyperspace.

Many modified PSO have been proposed and successfully applied to different optimization problems [5-12], such as a hybrid of GA and PSO (HGAPSO) [5], self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) [6], PSO with controller random exploration velocity (PSO-CREV) [7], a species-based PSO (SPSO) [8], and a hierarchical cluster-based multi-species particle swarm optimization (HCMSPSO) [12]. In HGAPSO and several latter studies [9-11], the idea of crossover in GA is incorporated into PSO. In SPSO and HCMSPSO, a swarm population is divided into species subpopulations, and the dominating particle in

each species is used as a best neighborhood for particles within the same species. This paper proposes a new framework for incorporating crossover operation and species formation into PSO.

This paper proposes a group-crossover PSO, called as GCPSO. According to the adaptively formed groups in a swarm, a new group-based crossover operation (GCRO) is proposed. Unlike the SPSO and HCMSPSO, the formation of the group is used not only for particle velocity update but also for particle crossover in the EGPSO. The EGPSO uses the name "group" because individuals in different species do not perform crossovers in nature. This paper applies the GCPSO algorithm to design a fuzzy controller for a truck reversing control problem. In the simulation, this paper demonstrates the performance of GCPSO via comparisons with various advanced GA and PSO algorithms.

This paper is organized as follows: Section II introduces the FC designed by the GCPSO. Section III introduces the GCPSO. Section IV conducts the truck reversing control problem. Finally, Section V presents conclusion.

II. FUZZY CONTROLLER(FC)

This paper designs a fuzzy controller (FC) using the GCPSO. The i th rule, denoted as R_i , in the FC is represented in the formula :

$$R_i : \text{If } x_1(k) \text{ is } A_{i1} \text{ And } \dots \text{ And } x_n(k) \text{ is } A_{in} \\ \text{Then } u(k) \text{ is } a_i \quad (1)$$

where k is time step, $x_1(k), \dots, x_n(k)$ are the input variables, $A_{i1}, \dots, A_{in}$ are fuzzy sets, and a_i is a crisp value. The fuzzy set A_{ij} uses a Gaussian membership function

$$M_{ij}(x_j) = \exp\left\{-\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right\}, \quad (2)$$

where m_{ij} and σ_{ij} represent the center and width of the fuzzy set A_{ij} , respectively. The fuzzy AND operation in the inference engine implemented by the algebraic product in the fuzzy theory. Therefore, given an input data set $\bar{x}(k) = (x_1(k), \dots, x_n(k))$, the firing strength $\phi_i(\bar{x}(k))$ is calculated by

$$\phi_i(\bar{x}(k)) = \prod_{j=1}^n M_{ij}(x_j(k)). \quad (3)$$

If there are r rules in the fuzzy controller, then the output of the controller computed by the weighted average defuzzification method and is shown

$$u(k) = \frac{\sum_{i=1}^r \phi_i(\bar{x}(k)) a_i}{\sum_{i=1}^r \phi_i(\bar{x}(k))}. \quad (4)$$

The designed of an FC training data consists of the determination of fuzzy rules and the optimization of all free parameters in each fuzzy rule. For the FC in (1) that consists of n inputs variables and r rules. Therefore, all of ties free parameters can be described by the following vector

$$\bar{s} = m_{11}, \sigma_{11}, \dots, m_{1n}, \sigma_{1n}, a_1, \dots, m_{r1}, \sigma_{r1}, \dots, m_{rn}, \sigma_{rn}, a_r \in \mathbb{R}^{r(2n+1)} \quad (5)$$

In GCPSO-designed FC, the number of rules r is assigned in advance. The number of fuzzy sets in each input variable is simply equal to the number of rules. The solution vector in (5) functions as a particle. The GCPSO-designed FC optimizes all of the free parameters in an FC with the number of rules being fixed.

III. GROUP-CROSSOVER PARTICLE SWARM OPTIMIZATION (GCPSO)

In basic PSO, the initial P_s (which is the swarm size) particles forming the swarm are randomly generated. However, there are too many search points before locating the global optimum. To improve the search ability, this paper applies the idea of swarm speciation proposed in species-based PSO (SPSO) [8] for particle velocity update. Unlike the SPSO, the formation of groups is used not only for particle velocity update but also for particle crossover in GCPSO. Figure 1 shows the block diagram of the group formation, particle update, particle generation, and replacement operations in the GCPSO. Details of these operations are introduced as follows.

A. Group formation

Initially, the particles are sorted from best performance to worst performance. The first particle is the best-performing one in the sorted swarm. This particle forms the first group and functions as first group leader (denoted as $\bar{p}_1^L$). For the leader $\bar{p}_z^L$ of group z , positions of the particles ranked after $\bar{p}_z^L$ are checked one by one against $\bar{p}_z^L$ until a particle whose

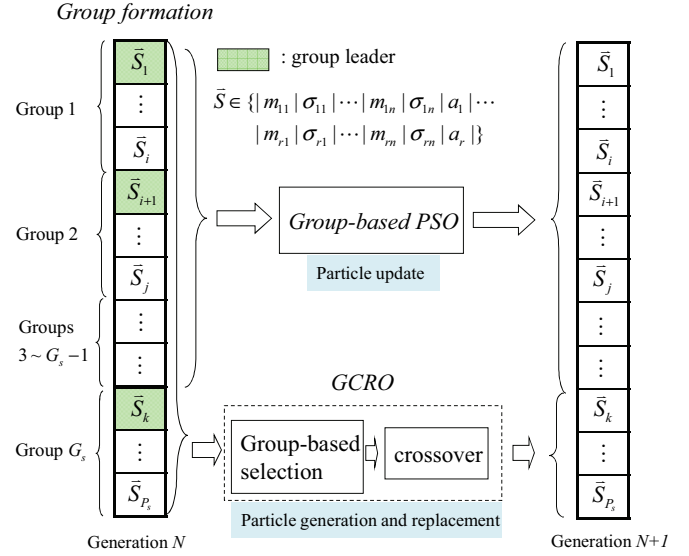


Fig. 1. The Block diagram of the GCPSO.

similarity degree with respect to $\bar{p}_z^L$ is larger than a predefined threshold ρ_{th} . The similarity degree, ρ , between a particle $\bar{s}_i$ and a group leader $\bar{p}_z^L$ is given by

$$\rho(\bar{p}_z^L, \bar{s}_i) = \sqrt{\frac{\|\bar{p}_z^L - \bar{s}_i\|_2^2}{D}}, \quad (6)$$

where D is the length of a particle vector. The ρ value can be regarded as the root-mean-squared error between two vectors. The particles with $\rho(\bar{p}_z^L, \bar{s}_i)$ being smaller than the threshold ρ_{th} are categorized into group z . The first particle with $\rho(\bar{p}_z^L, \bar{s}_i)$ being larger than the threshold ρ_{th} functions as a new leader $\bar{p}_{z+1}^L$ and forms a new group $z+1$. This paper sets the threshold ρ_{th} to be 1.5.

B. Particle Update

Suppose that there are G_s groups in the GCPSO, where G_s changes with learning iterations. The GCPSO updates particles in groups $1, \dots, (G_s - 1)$. In the GCPSO, grouping is used for neighborhood topology definition in a swarm. That is, the neighbors of a particle are defined as the particles within the same group. Therefore, the neighborhood best position is the leader $\bar{p}_z^L$ for particles within group z . The velocity $\bar{v}_i$ of each particle i in group z is updated using its individual best position, $\bar{p}_i$, and the neighborhood best position, $\bar{p}_z^L$. The following equation with a constriction coefficient is used for velocity updating [4]:

$$\begin{aligned} v_i^j(t+1) = & \chi(v_i^j(t) + c_1\phi_1^j \cdot (p_i^j(t) - s_i^j(t)) \\ & + c_2\phi_2^j \cdot (p_z^{lj}(t) - s_i^j(t))), \quad j = 1, \dots, D, \end{aligned} \quad (7)$$

where χ is a constriction factor controls the magnitude of $\bar{v}_i^j$, c_1 and c_2 are positive acceleration coefficients, and ϕ_i^j is uniform random number $[0, 1]$. The particles in different groups are attracted by the different group leaders, which causes different groups to locate different optima. This method helps to reduce the chance of getting stuck in a local solution. Each particle changes its position according to the following velocity-dependent formula:

$$\bar{s}_i(t+1) = \bar{s}_i(t) + \bar{v}_i(t+1). \quad (8)$$

C. Particle Generation and Replacement

The GCPSO generates new particles by using a new group-based crossover operation (GCRO), which is inspired by the crossover operation in the GAs. The GCRO-generated particles are used to replace the worst-performing group in each iteration. In the GCRO, parents for crossover are selected from different groups except for the worst-performing group in order to avoid inbreeding. Therefore, this operation is applied only when the number of groups G_s is larger than two. To select parents for the GCRO, the group-based tournament selection scheme is used. To select one parent, $G_s/2$ groups are first randomly and uniformly selected regardless of group performance. For these $G_s/2$ groups, the one with the best performance is selected. A particle within the selected group is then randomly and uniformly selected regardless of particle performance. This selected particle is designated as one parent $\bar{s}_{p1}$. The other parent $\bar{s}_{p2}$ is selected by the same way. Two new particles, $\bar{s}_{n1}$ and $\bar{s}_{n2}$, are created by performing crossover on the two selected parents. Two crossover sites are randomly selected and are denoted by t_1 and t_2 . The GCRO creates each element in the two new particle vectors using the following operations:

$$s_{n1}^j = \begin{cases} \gamma_1 s_{p1}^j + (1 - \gamma_1) s_{p2}^j, & j = 1, \dots, t_1 \\ \gamma_2 s_{p1}^j + (1 - \gamma_2) s_{p2}^j, & j = t_1 + 1, \dots, t_2 \\ \gamma_3 s_{p1}^j + (1 - \gamma_3) s_{p2}^j, & j = t_2 + 1, \dots, D \end{cases} \quad (9)$$

and

$$s_{n2}^j = \begin{cases} (1 - \gamma_1) s_{p1}^j + \gamma_1 s_{p2}^j, & j = 1, \dots, t_1 \\ (1 - \gamma_2) s_{p1}^j + \gamma_2 s_{p2}^j, & j = t_1 + 1, \dots, t_2 \\ (1 - \gamma_3) s_{p1}^j + \gamma_3 s_{p2}^j, & j = t_2 + 1, \dots, D \end{cases} \quad (10)$$

where γ_i is a uniform random number in $[0,1]$. Each crossover operation creates two new particles. For worst-group replacement, the number of crossover operations is equal to

Table 1. The SAE of GCPSO and Different Population-based Algorithms in for The Truck Reversing Control Problem.

Methods	EGA [1]	C-PSO [4]	HGAPSO [5]	HPSO-TVAC [6]	PSO-CREV [7]	GCPSO
Average	0.525	0.5438	0.141	0.3252	1.29	0.1204
STD	0.564	1.0643	0.176	0.257	0.67	0.1531

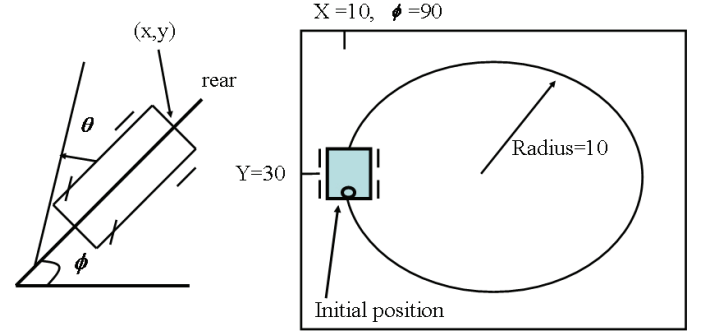


Fig. 2. Diagram of the simulation truck and the desired backing path.

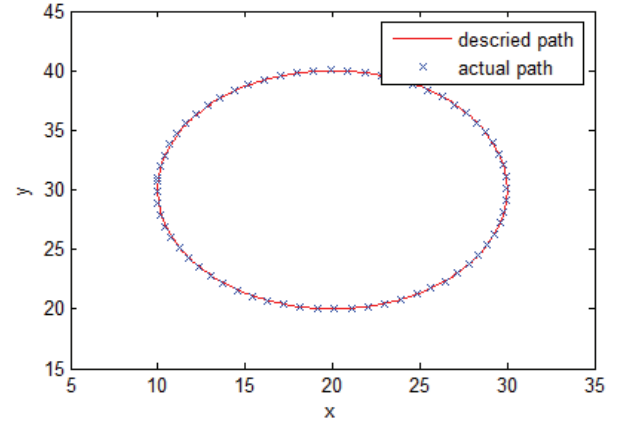


Fig. 3. The control results using GCPSO, where the desired and actual paths are denoted by solid line and "X", respectively.

$\lceil N_w / 2 \rceil$, where N_w denotes the number of particles in the worst-performing group.

For each new particle $\bar{s}_n$, its initial velocity $\bar{v}_n$ should be assigned accordingly, which is introduced as follows. The similarity degree between $\bar{s}_n$ and each of the left $G_s - 1$ group-leaders $\bar{p}_z^L$ are computed, and the leader with the minimum degree is found. That is, find

$$\hat{z} = \arg \min_{1 \leq z \leq G_s - 1} \rho(\bar{p}_z^L, \bar{s}_n) \quad (11)$$

The particle $\bar{s}_n$ is categorized into group $\hat{z}$. The initial velocity vector is assigned by the following formula:

$$\begin{aligned} v_n^j = & \gamma_4^j \cdot (p_z^{lj} - s_n^j) + \gamma_5^j \cdot (s_{p1}^j - s_n^j) \\ & + \gamma_6^j \cdot (s_{p2}^j - s_n^j), \quad j = 1, \dots, D \end{aligned} \quad (12)$$

This initial velocity is assigned according to the positions of the two parents and the group leader to which the new particle $\bar{s}_n$ belongs.

IV. SIMULATION

This paper simulates a truck reversing control problem. The FC is designed using GCP SO. Figure 2 shows the simulated truck. The truck position is precisely determined by the three state variables ϕ , x , and y , where ϕ is the angle of the truck with the horizon as shown in Fig. 2. The control signal to the truck is steering angle θ , where $\theta \in [-20^\circ, 20^\circ]$. Only reversing is considered. The truck moves backward by a fixed unit distance every stage. The model of the truck reversing system is [13]

$$x(k+1) = x(k) + \cos[\phi(k) + \theta(k)] + \sin[\theta(k)] \sin[\phi(k)] \quad (13)$$

$$y(k+1) = y(k) + \sin[\phi(k) + \theta(k)] - \sin[\theta(k)] \cos[\phi(k)] \quad (14)$$

$$\phi(k+1) = \phi(k) - \sin^{-1} \left[\frac{2 \sin(\theta(k))}{b} \right] \quad (15)$$

where b is the length of the truck and $b=4$ in this simulation. Equations (13) to (15) generate the next state when the present state and control are given. The task here is to design a fuzzy controller to reverse the truck following the desired circular path

$$(x-20)^2 + (y-30)^2 = 10^2 \quad (16)$$

with an initial state of $(x, y, \phi) = (10, 30, 90^\circ)$. Figure 3 shows the initial state and the desired driving path. There are a total of 65 control steps, and control error is defined as

$$SAE = \sum_{k=1}^{65} |e(k)|, e(k) = (((x(k)-20)^2 + (y(k)-30)^2)^{1/2} - 10)^2 \quad (17)$$

FC inputs are $x(k)$, $y(k)$, and $\phi(k)$, and the input output is θ . The number of fuzzy rules is set to be four. In the GCP SO-designed FC, the particle size is set to be 40. This paper sets the parameters in (7) as follows: $\chi = 0.9$, $C1 = C2 = 1$, and the $\rho_{th} = 1.5$. The number of iterations is 1000. For statistical analysis, learning is performed for 50 runs. Table 1 shows the average SAE and standard deviation (STD) over 50 runs.

For the sake of comparisons, this paper compares the GCP SO performance with different advanced GA and PSO, including: an elite GA (EGA)[1], constriction PSO (C-PSO) [4], a hybrid of GA and PSO (HGAPSO) [5], self-organizing hierarchical PSO with time-varying acceleration coefficients (HPSO-TVAC) [6], and PSO with controller random exploration velocity (PSO-CREV) [7]. The number of rules in these methods is also set to be four *a priori*. Table 1 shows the learning average SAE and STDs of these population-based optimization algorithms used for comparison. The result shows that the GCP SO achieves the smallest control error.

V. CONCLUSION

This paper proposes a new optimization algorithm, GCP SO, for truck reversing control problem. The GCP SO generates a new swarm by using not only group-based particle update but also group-based crossover operation. The group-based crossover operation (GCRO) is incorporated into PSO for performance improvement. Effectiveness and efficiency of the GCP SO have been demonstrated through comparisons with different population-based optimization algorithms in the FC design problem. In the future, the GCP SO-designed FC will be applied to other control problems to verify its performance.

REFERENCES

- [1] C.F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Trans. Fuzzy Systems*, vol.10, no. 2, pp. 155-170, April 2002.
- [2] C. F. Juang, "Combination of on-line clustering and Q-value based GA for reinforcement fuzzy system design," *IEEE Trans. Fuzzy Systems*, Vol. 13, No. 3, pp. 289-302, June 2005.
- [3] C. F. Juang and C. H. Hsu, "Temperature control by chip-implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm," *IEEE Trans. Circuits and Systems- I: Regular Papers*, Vol. 52, No. 11, pp. 2376-2384, Nov. 2005.
- [4] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, Feb. 2002.
- [5] C. F. Juang, "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, April 2004.
- [6] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evolutionary Computation*, vol. 8, no. 3, pp. 240-255, June 2004.
- [7] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Trans. Syst., Man, and Cyber., Part B: Cybernetics*, vol. 37, no. 5, pp. 1271-1289, Oct. 2007.
- [8] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Trans. Evolutionary Computation*, vol.10, no.4, pp.440-458, Aug. 2006.
- [9] Y. P. Chen, W. C. Peng, and M. C. Jian, "Particle swarm optimization with recombination and dynamic linkage discovery," *IEEE Trans. Syst., Man, and Cyber., Part B- Cyber.*, vol. 37, no. 6, pp. 1460-1470, Dec. 2007.
- [10] F. Grimaccia, M. Mussetta, and R.E. Zich, "Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Trans. Antennas and Propagation*, vol. 55, no. 3, Part 1, pp. 781-785, March 2007.
- [11] S. Jeong; S. Hasegawa, K. Shimoyama, and S. Obayashi, "Development and investigation of efficient GA/PSO-hybrid algorithm applicable to real-world design optimization," *IEEE Trans. Computational Intelligence Mag.*, vol. 4, no.3, pp. 36-44, Aug. 2009.
- [12] C. F. Juang, C. M. Hsiao, and C. H. Hsu, "Hierarchical cluster-based multi-species particle swarm optimization for fuzzy system optimization," *IEEE Trans. Fuzzy Systems*, vol. 18, no. 1, pp. 14-26, Feb. 2010..
- [13] L. X. Wang, *Adaptive Fuzzy Systems and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.