A Particle Swarm Optimization with Adaptive Multi-Swarm Strategy for Capacitated Vehicle Routing Problem

Kui-Ting Chen[1], Yijun Dai[2], Ke Fan[2] and Takaaki Baba[2]

Research Center[1] and Graduate School[2] of Information, Production and Systems, Waseda University

2-7 Hibikino, Kitakyushu, Fukuoka, Japan

nore@aoni.waseda.jp, lovingsaiyi@toki.waseda.jp, fanker0806@asagi.waseda.jp, tbaba@waseda.jp

***Abstract*— Capacitated vehicle routing problem with pickups and deliveries (CVRPPD) is one of the most challenging combinatorial optimization problems which include goods delivery/pickup optimization, vehicle number optimization, routing path optimization and transportation cost minimization. The conventional particle swarm optimization (PSO) is difficult to find an optimal solution of the CVRPPD due to its simple search strategy. A PSO with adaptive multi-swarm strategy (AMSPSO) is proposed to solve the CVRPPD in this paper. The proposed AMSPSO employs multiple PSO algorithms and an adaptive algorithm with punishment mechanism to search the optimal solution, which can deal with large-scale optimization problems. The simulation results prove that the proposed AMSPSO can solve the CVRPPD with the least number of vehicles and less transportation cost, simultaneously.**

Keywords- multi-swarm; particle swarm optimization; vehicle routing problem;adaptive algorithm

I. INTRODUCTION

Particle swarm optimization (PSO) is a powerful algorithm for finding an optimal solution in nonlinear search space. The PSO algorithms have been widely used in many applications. The main advantages of PSO algorithm are that it can produce excellent results with a reasonable resource cost and easy to be implemented in software [1]. However, the conventional PSO algorithm is difficult to be employed into combinatorial optimization problems such as capacitated vehicle routing problem with pickups and deliveries (CVRPPD) [2]. It includes several optimization subjects which are goods delivery/pickup optimization, vehicle number optimization, routing path optimization and transportation cost minimization. It is quite difficult for conventional PSO algorithm to find an optimal solution to simultaneously meet the requirements of different optimization subjects due to its simple search strategy.

Capacitated vehicle routing problem (CVRP) is one of the most challenging combinatorial optimization problems, which is introduced by G. B. Dantzig and J. H. Ramser in 1959 [3]. It concerns the problems of the goods distribution between depot and customers, which aims to simultaneously minimize the transportation cost and number of vehicles. The CVRPPD is an extension version of the classical CVRP, where customers may both receive and send goods with a fixed capacity of vehicles. In the CVRPPD, the combination of a possible solution set is much more than the CVRP, since the pickup derive has a huge impact on the routing optimization. For example, if the quantity of both the pickup and delivery is required 20, the maximum capacity of each vehicle is 100. As shown in Fig. 1, a purple routing path is a classic solution for CVRP, in which the vehicle can deliver the goods to customers without exceed the maximum capacity. In contrast, a red routing path is an impossible routing for the CVRP, since the total quantity (120) of required goods is over the capacity (100) of the vehicle for six customers. However, if a pickup service is required in the red routing path, it will become a possible solution even if there are seven customers. The pickup service drastically increases the number of the possible solutions. It becomes much more difficult to find the optimal solution in the CVRPPD.

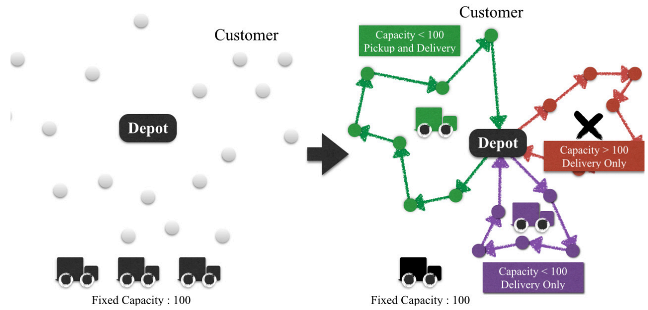

Figure 1. Concept of CVRPPD.

In order to overcome the above difficulty, a PSO algorithm with adaptive multi-swarm strategy (AMSPSO) is proposed. It can provide an adaptive search behavior for dealing with large-scale optimization problems. The proposed approach divides a particle swarm into various small groups which cooperate with an adaptive algorithm. The each group of swarm employs different PSO algorithms which can provide different search ability such as global search ability, local search ability and so on. The proposed approach exploits the adaptive algorithm to

INISCom 2015, March 02-04, Tokyo, Japan

DOI 10.4108/icst.iniscom.2015.258972

regulate the number of the swarm groups according to the current convergence status of the whole particle swarm, which can immediately optimize search strategy for PSO algorithm.

The rest of this paper is organized as follows. In section 2, the concept of PSO algorithm and CVRPPD are briefly introduced. In section 3, the details of the proposed multi-swarm strategy of PSO algorithm is presented. In section 4, the simulation results of the proposed and conventional approach are provided. Finally, section 5 comprises a summary and the conclusions of this research.

II. Related Works

A. Vehicle Routing Problem

In the definition of CVRPPD [2], every vehicle (k) has a fixed cost of f, variable cost per distance unit g, capacity Q, and service duration limit D. Each customer (i) has a non-negative pickup quantity p_i, delivery quantity q_i, and a service time s_i. The optimal solution of the CVRPPD is a set of m routes, which must meet the requirement as follows

- Each route starts and ends at the depot.
- Each customer (i) is visited once by one vehicle (k).
- The total load vehicle in any path does not exceed the capacity of the vehicle assigned to it (Q).
- The total transportation time of each vehicle does not exceed a preset limit D.
- The total cost (Z) is minimized.

The formulation of CVRPPD is given by [2]:

$$\text{Minimize } Z = f\sum_{k=1}^{m}\sum_{j=1}^{n} x_{0jk} + g\sum_{i=0}^{n}\sum_{j=1}^{n+1}\sum_{k=1}^{m} d_{ij}x_{ijk} \quad (1)$$

Subject to

$$\sum_{i=0}^{n}\sum_{k=1}^{m} x_{ijk} = 1 \quad \text{for } 1 \le j \le n \quad (2)$$

$$\sum_{j=0}^{n} x_{ijk} = \sum_{j=1}^{m} x_{ijk} \quad \text{for } 1 \le i \le n,\ 1 \le k \le m \quad (3)$$

$$\sum_{j=1}^{n} x_{0jk} \le 1 \quad \text{for } 1 \le k \le m \quad (4)$$

$$\delta_{ik} + s_i + t_{ij} - \delta_{jk} \le (1 - x_{ijk})M \quad \text{for } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \quad (5)$$

$$\delta_{n+1,k} - \delta_{0k} \le D \quad \text{for } 1 \le k \le m \quad (6)$$

$$y_{ijk} \le x_{ijk}Q \quad \text{for } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \quad (7)$$

$$\sum_{j=1}^{n} y_{0jk} = \sum_{j=1}^{n} q_j \sum_{i=0}^{n} x_{ijk} \quad \text{for } 1 \le k \le m \quad (8)$$

$$\sum_{i=0}^{n} y_{ijk} + (p_j - q_j)\sum_{i=0}^{n} x_{ijk} = \sum_{i=1}^{n+1} y_{ijk} \quad \text{for } 1 \le j \le n,\ 1 \le k \le m \quad (9)$$

$$x_{ijk} \in \{0,1\} \quad \text{for } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \quad (10)$$

$$y_{ijk} \ge 0 \quad \text{for } 0 \le i \le n,\ 1 \le j \le n+1,\ 1 \le k \le m \quad (11)$$

$$\delta_{ik} \ge 0 \quad \text{for } 0 \le i \le n+1,\ 1 \le k \le m \quad (12)$$

where n is the total number of the customers. m is the number of the total routing paths. x_{ijk} represents that a binary variable indicating status of each path (i, j) is traversed by vehicle k. y_{ijk} is load capability of vehicle k while traversing path (i, j). δ_{ik} is starting service time of customer i by vehicle k. d_{ij} and t_{ij} are a distance matrix and a travel time matrix, respectively. Equation (1) minimizes routing cost, which consists of transportation fixed cost and variable cost. Equations (2) and (3) ensure that every customer is visited by one vehicle exactly. Equations (5) and (6) define the relationship between service time (s_i) and travel time (t_{ij}). The total transpiration time of vehicle cannot exceed the duration limit D. Vehicle load constraints are explained in (7), (8) and (9). Each vehicle cannot over load the goods during the pickup and deliver. Equations (10), (11) and (12) state the domain of decision variables: all x_{ijk} are binary variables, y_{ijk} and δ_{ik} are positive real variables [2].

B. PSO Algorithm for Vehicle Routing Problem

PSO is a stochastic optimization algorithm based on swarm intelligence, which was introduced by J. Kennedy and R. Eberhart in 1995 [4]. The basic operation of PSO algorithm is updating the position and velocity of particle to find an optimal solution. Each particle l has current velocity v_l and a personal best position p_{ld} which represents a possible solution of optimization space. Considering an d-dimensional evaluation function, the position and velocity of the particle l in $(t+1)^{th}$ iteration are updated by the following equations:

$$v_{ld}^{t+1} = \omega * v_{ld}^{t} + c_1 * r_1 * (p_{ld} - x_{ld}^{t}) + c_2 * r_2 * (p_{gd} - x_{ld}^{t}) \quad (13)$$

$$x_{ld}^{t+1} = x_{ld}^{t} + v_{ld}^{t+1} \quad (14)$$

where r_1 and r_2 are uniformly random numbers in the range [0,1], p_{gd} is the location of the particle when the best fitness value is obtained for the whole population, c_1 and c_2 are two acceleration constants, ω is called the inertia weight factor, and d is the number of dimensions in the search space.

In the conventional PSO algorithm, the position and velocity of particle are defined in (13) and (14), respectively. The values of position and velocity are represented by real number. However, most variables of the CVRPPD are represented by binary number as mentioned in previous section. In order to employ PSO algorithm into CVRPPD, the real number needs to encode/decode for representing the binary

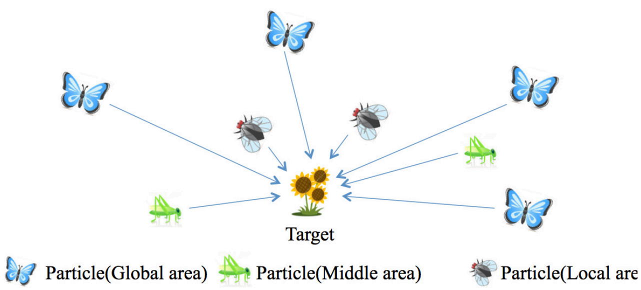

Figure 2. Concept of AMSPSO.

variables. Some encoding/decoding approaches are introduced in [5, 6].

T. J. Ai and V. Kachitvichyanukul proposed two different encoding/decoding approaches that are named SR-1 and SR-2 [5]. It provides the approaches to transform the position and velocity of particle from real number to binary number. In the SR-1, they increased the dimension number of particle to represent *n* customers and *m* vehicles. The dimension number of particles is defined by (*n*+*2m*). In the SR-2, they transform a particle into the vehicle orientation points and the vehicle coverage radius. The dimension number of particles is defined by (*3m*). Their simulation results proved that SR-2 can produce better result than SR-1, since SR-1 leads to a larger number of particle's dimension than SR-2. In the comparison of calculation speed, the calculation speed of the SR-1 is much faster than SR-2. In addition, SR-1 is more suitable for dealing with CVRPPD, since SR-2 is difficult to take the requirements of customers into encoding/decoding procedure. However, it is difficult for the conventional PSO algorithm to find the optimal solution under (*n*+*2m*) dimension search space. In order to overcome this difficulty, the AMSPSO is proposed. The SR-1 is also employed in the proposed AMSPSO.

III. PSO WITH ADAPTIVE MULTI-SWARM STRATEGY

The proposed MSPSO divides particles into several groups, as illustrated in Fig. 2. Each group employs the different PSO algorithms, which can maintain global search ability and local search ability. In addition, the search behavior of proposed algorithm is more similar to human society.

As shown in the Fig. 2, the particles are divided into three groups as an example. One group is expert in the searching optimization solution on a global area, which employs quantum-behaved PSO (QPSO). Li *et al.* proved that QPSO is powerful on searching the optimal solution even if it is applied into a high dimensional search space [7]. The second group employs a PSO with random time-varying inertia weight and acceleration coefficients (PSO-RTVIWAC) which has a powerful searching ability on a local area [8]. The third one is PSO with passive congregation (PSOPC) which can help individuals to avoid misjudging information and becoming trapped by poor local optimal solution [9]. By employing the above PSO algorithms into different groups, the proposed approach cannot only prevent particles from converging on a local optimal solution, but also achieve powerful search ability on global and local area.

In this paper, different PSO algorithms are combined into generic equations based on the method which is introduced in [1]. The generic equations are given by

$$\begin{aligned} v_{id}^{t+1} &= sel_1 * [\omega * v_{id}^t + c_1 * r_1 * (p_{id} - x_{id}^t) \\ &\quad + c_2 * r_2 * (p_{gd} - x_{id}^t) \\ &\quad + sel_2 * c_3 * r_3 * (R_i^t - x_{id}^t)] \end{aligned} \tag{15}$$

$$\begin{aligned} x_{id}^{t+1} &= sel_3 * x_{id}^t + sel_4 * v_{id}^{t+1} + sel_5 * p_{id} + sel_6 * p_{gd} \\ &\quad \pm sel_7 * \beta * \left| mbest - x_{id}^t \right| * \ln\left(\frac{1}{r_4}\right) \end{aligned} \tag{16}$$

$$mbest = \sum_{i=1}^{N} \frac{p_{id}}{\mathrm{N}} \tag{17}$$

where sel_1, sel_2, sel_3, sel_4, sel_5, sel_6 and sel_7 are the particle motion coefficients, other parameters have been defined in [1]. The type of PSO algorithm can be changed by setting the values of particle motion coefficients, as presented in Table 1. The generic equation did not define the parameters (ω, c_1, c_2 and c_3) of PSO algorithm. In order to provide the better search performance of PSO algorithm, new calculation equation of ω, c_1, c_2 and c_3 are given by

$$\omega = sel_8 * r_5 * (\omega_{\max} - t * (\omega_{\max} - \omega_{\min}) / T) + sel_9 * \frac{r_6}{2} + sel_{10} \tag{18}$$

$$c_1 = r_7 * \left(c_{1\max} - t * (c_{1\max} - c_{1\min}) / T\right) + sel_{11} \tag{19}$$

$$c_2 = r_8 * \left(c_{2\max} - t * (c_{2\max} - c_{2\min}) / T\right) + sel_{12} \tag{20}$$

TABLE I. THE EXAMPLES OF PARTICLE MOTION COEFFICIENT FOR CHANGING TYPE OF PSO

Type of PSO Algorithm	**Particle Mention Coefficients**												
	sel_1	sel_2	sel_3	sel_4	sel_5	sel_6	sel_7	sel_8	sel_9	sel_{10}	sel_{11}	sel_{12}	sel_{13}
Original PSO [4]	1	0	1	1	0	0	0	0	0	1	2	2	0
PSO-RTVIWAC [8]	1	0	1	*	0	0	0	1	0	0	0	0	0
QPSO [7]	0	0	0	0	*rand*	1-sel_5	1	0	0	0	0	0	0
PSOPC [9]	1	1	1	1	0	0	0	1	0	0	0.5	0.5	*
Standard PSO[10]	*cst.*	0	1	1	0	0	0	0	0	0	0	0	0
PSO-TVIW [11]	1	0	1	1	0	0	0	1	0	0.4	2	2	0
PSO-TVAC [12]	1	0	1	1	0	0	0	0	0	0.9	c_{1min}	c_{2max}	0
PSO-RANDIW [13]	1	0	1	1	0	0	0	0	1	0.5	1.49	1.49	0
Gaussian PSO [14]	1	0	1	1	0	0	0	0	0	*cst.*	*cst.*	*cst.*	0

*:The value of the particle motion coefficient changes dynamically. *rand*: A uniform random number. *cst.*: A constant value.

$$c_3 = r_9 * \left(c_{3\max} - t * (c_{3\max} - c_{3\min}) / T\right) + sel_{13} \quad (21)$$

where t is the current iteration times, T is the maximal iteration times. The examples of particle motion coefficient for changing type of PSO are shown in Table I.

A. Adaptive Multi-Swarm Strategy with Punishment Mechanism

In the proposed approach, the particle swarm is divided into three groups to maintain the global search and local search ability. However, the particle number of each group cannot be a fixed value, since the global search ability has a huge impact on the early stage of the iterations. In contrast situation, the local search ability plays an important role during the later stage. Therefore, an appropriate regulation of the particle number can drastically improve the performance of the proposed approach.

In order to figure out the appropriate regulation, an adaptive algorithm with punishment mechanism is proposed in this section. The adaptive algorithm aims to find the best combination of the particle numbers for each group. It exploits the punishment mechanism to arbitrate the all of the swarm groups for the current convergence status. Meanwhile, the punishment mechanism increases/decreases particle number of the swarm groups. The punishment mechanism makes swarm groups compete with each other, which is like resource plunder in human society. The winner can plunder most resources in the whole society. It means that the particle number of each swarm group is going to be increased or decreased which is based on its search performance. The search performance of all swarm groups has to be evaluated until all iterations is finished. In the beginning of the iterations, the punishment mechanism assigns the same particle number to each swarm group with a same credibility which is used for evaluating its search performance. The higher credibility can win more number of particles from other groups to assign into its swarm group. The credibility of each swarm group is a counting value when *gbest* is updated by the own particles. The equation of credibility (*Credi*) is given by

$$Credi_{\varphi}^{t+1} = Credi_{\varphi}^{t} + t * reward + 1 \quad (22)$$

where φ is the number of swarm group. t is the current iteration times. *reward* is an additional reward for updating the *gbest* during the iteration. The additional reward is used to encourage the swarm group when it can produce better *gbest* during searching procedure. The punishment mechanism ranks the credibility of each swarm group with a fixed iteration cycle named punishment cycles. For example, the punishment mechanism calculates the total credibility of three swarm groups at each 25 iteration times. The total particle number of whole groups is 50. The ranking credibility at first place (*Group 1*) can assign 25 particles into its group. The second place (*Group 2*) can take 15 particles. The left 10 particles are for third group (*Group 3*). Considering the *gbest* is very easier to be updated during the early iterations, the additional reward is proportional to the number of iterations. However, it still cannot stop the *Group 1* to rapidly accumulate the credibility in the early iterations. It leads to *Group 3* never win the first place of the ranking credibility. In the punishment cycles, the value of the credibility is reassessed by

$$Credi_{\varphi}^{t+1} = Credi_{\varphi}^{t} * \left(1 - \frac{P_{\varphi}}{P_{total}}\right) \quad (23)$$

where P_{φ} is assigned particle number of its swarm group. P_{total} is the total number of all swarm groups. Equation (23) can drastically decrease the credibility of the winner group when its search performance is not good enough. The proposed AMSPSO employs the punishment mechanism which can regulate the search strategy with considering the convergence status of all particles. The above proposed approaches are evaluated in CVRPPD.

IV. Simulation Results

The proposed AMSPSO algorithm is implemented by C# language with using Microsoft Visual Studio 2010 on a PC with Intel Core i3 3.2 GHz and 4 GB RAM. The required parameters of the simulation are shown in Table II. Seven sets of benchmark instance data (CMT1T to CMT5T and CMT11T to CMT12T) are prepared [2]. The parameters of each benchmark instance are shown in Table III. In the CMT1T to CMT5T and CMT11T to CMT12T, the vehicle can deliver/pickup the goods to customers. Both of the fixed cost (f) and cost per distance unit (g) is set as 1. Each benchmark instance is executed 1,000 runs with 50 particles and 500 iteration times. The AMSPSO is evaluated by the above benchmark instances and compared with the conventional PSO algorithm [2]. The all of the simulation environments are set same with [2]. The improve ratio (IR) is defined by

$$IR(\%) = \frac{Z^* - Z}{Z} * 100\% \quad (24)$$

TABLE II. Summary of Simulation Parameters

Parameters	Values
Number of particle	50
Number of iteration	500
Punishment cycles	50 iterations
Type of PSO algorithm	QPSO, PSO-RTVIWAC, PSOPC
PSO parameters	ω = 0.4 to 0.9 β = 0.4 to 0.9 c_{min} = 0.5 c_{max} = 2.5 *reward* = 0.004
Particle motion coefficient (sel_4)	0.171 to 1.0 (PSO-RTVIWAC)
Particle motion coefficient (sel_{13})	0.4 to 0.6 (PSOPC)

TABLE III. Parameters of CMT

Benchmark Instances	Customer Numbers	Capacity of Vehicle (Q)
CMT1T	50	160
CMT2T	75	140
CMT3T	100	200
CMT4T	150	200
CMT5T	199	200
CMT11T	120	200
CMT12T	100	200

TABLE IV. SIMULATION RESULTS

Benchmark Instances	Customer Numbers	Best Solution of Conventional PSO [2]		Best Solution of AMSPSO		Improved Ratio (*IR*)
		No. of Vehicles	*Total Cost (Z)*	*No. of Vehicles*	*Total Cost (Z*)*	
CMT1T	50	5	520	5	520	0.00%
CMT2T	75	9	810	9	**793**	**2.09%**
CMT3T	100	7	827	7	**807**	**2.42%**
CMT4T	150	11	1014	11	1014	0.00%
CMT5T	199	15	1297	15	**1296**	**0.08%**
CMT11T	120	7	**1026**	7	1027	-0.09%
CMT12T	100	9	792	9	**788**	**0.51%**

where Z^* is the total cost of AMSPSO. Z is the total cost of the conventional PSO algorithm.

The simulation results of the benchmark instance are shown in Table IV. In most of the benchmark instances, the performance of the AMSPSO can produce a better results compared with the conventional PSO algorithm, even if the customer size is increased. In the instance of CMT2T and CMT3T, the AMSPSO can respectively reduce the total cost by 17 and 20 within 500 iteration times which is 50% of the conventional approach. In addition, the proposed AMSPSO can reduce the total cost by 2.42% and 2.09% in CMT3T and CMT2T respectively, as shown in Table III. The simulation results prove that the proposed AMSPSO can realize the less cost than conventional PSO algorithm [2]. The conventional PSO algorithm needs 1,000 iteration times to achieve the same level results. It proves that proposed AMSPSO can solve the VRPPD with less cost than conventional PSO algorithm. In addition, some new best known solutions of the benchmark instances are also found by the proposed MSPSO. As the future work, the AMSPSO will compare with others similar approaches to further evaluate the performance under different kinds of VRP instances.

V. CONCLUSIONS

In this paper, a particle swarm optimization with adaptive multi-swarm strategy (AMSPSO) is proposed to solve a capacitated vehicle routing problem with pickups and deliveries (CVRPPD). The proposed AMSPSO employs the multiple PSO algorithms and an adaptive algorithm with punishment mechanism. The multiple PSO algorithms can simultaneously maintain the global and local search ability. The adaptive algorithm with punishment mechanism can drastically improve the performance of the multi-swarm strategy to reduce the iteration times. The proposed approaches can dynamically regulate the search strategy for dealing with large-scale optimization applications. The simulation results prove that the proposed AMSPSO cannot only reduce 50% iteration times of the conventional approach, but also reduce the transportation cost by 2.42% at most. The AMSPSO can solve the CVRPPD with the less transportation cost. In addition, some new best known solutions of the benchmark problem are also found by the proposed AMSPSO. As the future work, the more complex benchmark instances of the CVRPPD will be prepared for further evaluating the performance of the proposed approach.

ACKNOWLEDGMENT

This work was supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (No. 26280017). The authors thank Yang Zeng, Ying Deng, Qi Liang, and Yu Zhang for their efforts to do the paper work in a very restricted timeframe during the paper preparation.

REFERENCES

[1] K.-T. Chen, M. Jia, J. Zhang, and T. Baba, "A flexible hardware architecture for particle swarm optimization." Journal of Signal Processing, Vol. 16, No. 6, pp. 519-526, Nov. 2012.

[2] T. J. Ai and V. Kachitvichyanukul, "A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery," Computers & Operations Research, Vol. 36, Issue 5, pp. 1693-1702, May 2009.

[3] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," Management Science, Vol. 6, Issue 1, pp. 80-91, Oct. 1959.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings IEEE Int. Conf. on Neural Networks, pp. 1942-1948, Nov. 1995.

[5] T. J. Ai and V. Kachitvichyanukul, "Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem," Computers & Industrial Engineering, Vol. 56, Issue 1, pp. 380-387, Feb. 2009.

[6] B. Wu, W. Wang, Y. Zhao, X. Xu, and F. Yang, "A novel real number encoding method of particle swarm optimization for vehicle routing problem," 6th World Congress on Intelligent Control and Automation, pp. 3271-3275, June 2006.

[7] J. Liu, J. Sun, and W. Xu, "Improving quantum-behaved particle swarm optimization by simulated annealing," Proceedings Computational Intelligence and Bioinformatics, Lecture Notes in Computer Science, LNBI 4115, pp. 130-136, Aug. 2006.

[8] H. Zhu, Y. Tanabe, and T. Baba, "A Random time-varying particle swarm optimization for the real time location systems," IEEJ Trans. on Electronics, Information and Systems, Vol. 128-C, No. 12, pp. 1747-1760, Nov. 2008.

[9] S. He, Q. H. Wu, J. Y. Wen, J. R. Saunders, and R.C. Paton, "A particle swarm optimization with passive congregation," BioSystems, Vol. 78, No. 1-3, pp. 135-147, Dec. 2004.

[10] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," Proceedings IEEE Swarm Intelligence Symposium, pp. 120-127, 2007.

[11] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," Proceedings IEEE Int. Conf. on Evolutionary Computation, Vol. 3, pp. 101-106, Apr. 1999.

[12] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," IEEE Trans. on Evolutionary Computation, Vol. 8, No. 3, pp. 240-255, June 2004.

[13] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," Proceedings IEEE Congress on Evolutionary, Computation, pp. 94-97, May 2001.

[14] R. A. Krohling, "Gaussian swarm: a novel particle swarm optimization algorithm," Proceedings IEEE International Conference on Cybernetics and Intelligent Systems, pp. 372-376, Dec. 2004.