EAI Endorsed Transactions Preprint

Research Article

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

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

[1]Research Center and [2]Graduate School of Information, Production and Systems, Waseda University, 2-7 Hibikino, Kitakyushu, Fukuoka, Japan

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.

Received on 31 March 2015; accepted on 31 March 2015; published on XXXX

Keywords: multi-swarm, particle swarm optimization, vehicle routing problem, adaptive algorithm

doi:10.4108/XX.X.X.XX

1. Introduction

Particle swarm optimization (PSO) is a powerful algorithm for finding an optimal solution in nonlinear search space. The PSO algorithm has 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 was introduced by G. B. Dantzig and J. H. Ramser in 1959 [3]. It concerns the problem of the goods distribution between depot and customers, which aims to simultaneously minimize the transportation cost and the 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 Figure 1, a purple routing path is a classic solution for CVRP, in which the vehicle can deliver the goods to customers without exceeding the maximum capacity. In contrast, a red routing path is an impossible route for the CVRP, since the total quantity (120) of required

★This is the extended version of the paper published in INISCom 2015

*Corresponding author. Email: nore@aoni.waseda.jp

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.

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 abilities such as global search ability, local search ability and so on. The proposed approach exploits the adaptive algorithm to 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. In section 5, the contributions of the AMSPSO for existing industrial applications are discussed. Finally, section 6 comprises a summary and the conclusions of this research.

2. Related Works

2.1. 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

(i) Each route starts and ends at the depot.

(iii) Each customer (i) is visited once by one vehicle (k).

(iii) The total load of vehicles does not exceed the capacity (Q) during the deliver and pickup.

(iv) The total transportation time of each vehicle does not exceed a service duration limit D.

(vi) The total cost (Z) is minimized.

The formulation of CVRPPD is given by [2]:

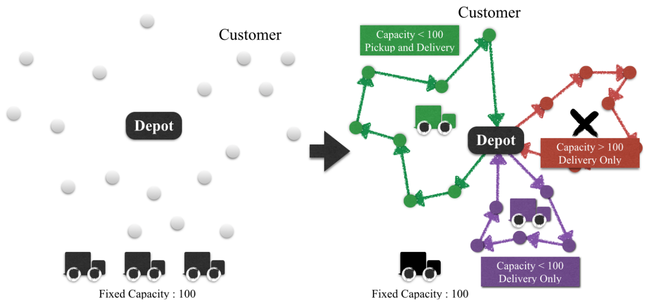

Figure 1. Concept of CVRPPD

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

Subject to

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

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

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

$$\begin{gathered} \delta_{ik} + s_i + t_i j - \delta_{jk} \leq (1 - x_{ijk})M \\ \text{for } 0 \leq i \leq n, 1 \leq j \leq n+1, 1 \leq k \leq m \end{gathered} \quad (5)$$

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

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

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

$$\sum_{i=0}^{n} y_{ijk} + (p_j - q_j) \sum_{i=0}^{n} x_{ijk} = \sum_{i=1}^{n+1} y_{ijk} \quad (9)$$

$$\text{for } 1 \leq j \leq n, 1 \leq k \leq m$$

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

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

$$\delta_{ik} \geq 0 \quad \text{for } 1 \leq j \leq n+1, 1 \leq k \leq m \quad (12)$$

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). $_{ik}$ 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 $_{ik}$ are positive real variables [2].

2.2. 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:

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

$$x_{ld}^{t+1} = v_{ld}^{t+1} + x_{ld}^{t} \tag{14}$$

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 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]. These two approaches 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

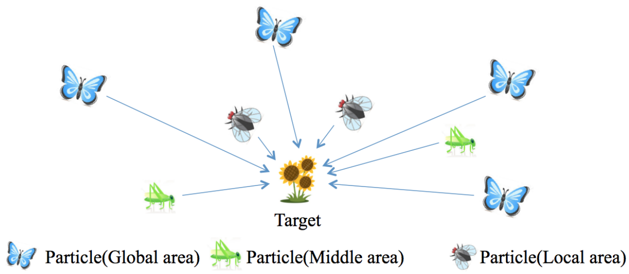

Figure 2. Concept of the proposed PSO with multi-swarm

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.

3. PSO with Adaptive Multi-Swarm Strategy

The proposed MSPSO divides particles into several groups, as illustrated in Figure 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.

3.1. Multi-swarm strategy with mixed PSO

As shown in the Figure 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

Table 1. The examples of particle motion coefficient for changing type of PSO

Type of PSO algorithm	Particle Motion 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 [0,1].
$cst.$:A constant value.

introduced in [1]. The generic equations are given by

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

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

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

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]. N is the population size of the particle swarm and $mbest$ is mean of the personal best position of all particles. 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, β, 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, β, c_1, c_2 and c_3 are given by:

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

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

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

$$c_3 = r_9 * (c_{3max} - \frac{t * (c_{3max} - c_{3min})}{T}) + sel_{13} \tag{21}$$

where t is the current iteration times, T is the maximal iteration times. The examples of particle motion coefficient for changing type of PSO algorithm are shown in Table 1.

3.2. Adaptive multi-swarm strategy

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, 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.

Punishment mechanism. 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 all 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.

Search performance evolution with credibility. The credibility of each swarm group is a counting value when global best(p_{gd}) is updated by the own particles. The equation of credibility ($Credi$) is given by

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

where φ is the number of swarm group. t is the current iteration times. $reward$ is an additional reward for updating the global best during the iteration. The additional reward is used to encourage the swarm group when it can produce better global best during searching procedure. Considering the global best is very easy to be updated during the early iterations, the additional reward is proportional to the number of iterations.

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 credibility of each 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 2 particles into its group. The particle number of the second place (*Group 2*) is not changed. The particle number of the third group (*Group 3*) is deceased 2 particles. To prevent number of the swarm groups decrease to zero, the particle number of each group must to keep a fixed minimum value (P_{min}). Once the particle number of third group reaches P_{min}, *Group 1* can get 2 particles from the *Group 2*. After the particle number of each swarm groups is reassigned, the credibility of each swarm group is reassessed by punishment mechanism to avoid that one group possess a great number of particles.

Credibility reassessing. In the (22), the additional reward is proportional to the number of iterations. However, it still cannot stop the Group A to rapidly accumulate the credibility in the early iterations. It leads to Group C never win the first place of the ranking credibility. In the punishment cycles, the value of the credibility is reassessed by

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

where P_{φ} is assigned particle number of its swarm group. P_{total} is the population size of whole 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.

4. Simulation Results

The proposed AMSPSO algorithm is implemented by C# language with using Microsoft Visual Studio 2013 (.Net Framework 4.5) on a PC with Intel Core i7 3.6 GHz and 32 GB RAM. Three sets of benchmark instance data (CMT1 to CMT14) which are named CMTnT, CMTnQ and CMTnH [15]. The pickup ratio of the three sets is referred to 10% (CMTnT), 25% (CMTnQ) and 50% (CMTnH). In our previous research [16, 17], a performance analysis has been carried out by using CMTnT. In this paper, some parameters are changed to further evaluate the performance of the proposed AMSPSO. The required parameters of the simulation are shown in Table 2. The parameters of each benchmark instance are shown in Table 3.

In the CMT1 to CMT5 and CMT11 to CMT12, the vehicle can deliver/pickup the goods to customers without considering the service time limitation during the transportation, since the transportation time of the

Table 2. Summary of simulation parameters

Parameters	Values
Number of particle	50
Number of iteration	500
Punishment cycles	25 iterations
Type of PSO algorithm	QPSO, PSO-RTVIWAC, PSOPC
PSO paramenters	ω = 0.4 to 0.9, β = 0.4 to 0.9, c_{min} = 0.5, c_{max} = 2.5, $reward$ = 0.004, P_{min} = 10
Particle motion coefficient (sel_4)	0.171 to 1.0 (PSO-RTVIWAC)

Table 3. Parameters of CMT instances

Instances (T, Q, H)	Capacity of Vehicle (Q)	Service Time Limit (D)	Service Time (s_i)
CMT1	160	∞	0
CMT2	140	∞	0
CMT3	200	∞	0
CMT4	200	∞	0
CMT5	200	∞	0
CMT6	160	200	10
CMT7	140	160	10
CMT8	200	230	10
CMT9	200	200	10
CMT10	200	200	10
CMT11	200	∞	0
CMT12	200	∞	0
CMT13	200	720	50
CMT14	200	1040	90

Table 4. Simulation results of CMTnT

Benchmark Instances	Customer Numbers	Best Solution of Conventional PSO [2] No. of Vehicles	Total Cost (Z)	Best Solution of AMSPSO No. of Vehicles	Total Cost (Z^*)	Improve Ratio (%)
CMT1T	50	5	520	5	520	0.00%
CMT2T	75	9	810	9	**794**	**1.98%**
CMT3T	100	7	827	7	**807**	**2.54%**
CMT4T	150	11	1014	11	1014	0.00%
CMT5T	199	15	1297	15	**1296**	**0.08%**
CMT6T	50	6	555	6	555	0.00%
CMT7T	75	12	942	**11**	**914**	**2.97%**
CMT8T	100	9	904	9	**876**	**3.10%**
CMT9T	150	14	1206	14	**1201**	**0.41%**
CMT10T	199	18	1502	18	**1470**	**2.13%**
CMT11T	120	7	1026	7	1027	-0.10%
CMT12T	100	9	792	9	**788**	**0.51%**
CMT13T	120	11	1548	11	1556	-0.52%
CMT14T	100	10	846	10	848	-0.24%
Average Improve Ratio						**0.92%**

Table 5. Simulation results of CMTnQ

Benchmark Instances	Customer Numbers	Best Solution of Conventional PSO [2] No. of Vehicles	Total Cost (Z)	Best Solution of AMSPSO No. of Vehicles	Total Cost (Z^*)	Improve Ratio (%)
CMT1Q	50	4	490	4	**489**	**0.20%**
CMT2Q	75	8	739	8	**734**	**0.68%**
CMT3Q	100	6	768	6	**753**	**1.95%**
CMT4Q	150	9	938	9	**921**	**1.81%**
CMT5Q	199	13	1174	13	**1162**	**1.02%**
CMT6Q	50	6	557	6	**555**	**0.36%**
CMT7Q	75	12	933	**11**	**904**	**3.11%**
CMT8Q	100	9	890	9	**869**	**2.36%**
CMT9Q	150	14	1214	14	**1191**	**1.89%**
CMT10Q	199	18	1509	18	**1444**	**4.31%**
CMT11Q	120	6	964	6	972	-0.83%
CMT12Q	100	7	733	7	**730**	**0.41%**
CMT13Q	120	11	1570	11	**1556**	**0.89%**
CMT14Q	100	10	825	10	**821**	**0.48%**
Average Improve Ratio						**1.33%**

vehicle is infinite. In the CMT6 to CMT10 and CMT13 to CMT14, the transportation time of the vehicle is limited. The vehicles have to finish the deliver/pickup and return to the depot within the service duration limit (D), as shown in Table 3. In addition, each vehicle will spent 10 (CMT6 to CMT10), 50 (CMT13) or 90 (CMT14) limitation time to service a customer. Both of the fixed cost (f) and cost per distance unit (g) is set as 0 and 1, respectively. Each benchmark instance is executed 10 runs with 50 particles and 500 iteration times. The particle number of QPSO, PSO-RTVIWAC and PSOPC is set by 17, 17 and 16, respectively. 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

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

where Z^* is the total cost of AMSPSO. Z is the total cost of the conventional PSO algorithm. The simulation results are shown in Table 4, 5 and 6.

Table 6. Simulation results of CMTnH

Benchmark Instances	Customer Numbers	Best Solution of Conventional PSO [2] No. of Vehicles	Total Cost (Z)	Best Solution of AMSPSO No. of Vehicles	Total Cost (Z^*)	Improve Ratio (%)
CMT1H	50	3	464	3	**461**	**0.65%**
CMT2H	75	6	668	6	**661**	**1.05%**
CMT3H	100	4	701	4	**697**	**0.57%**
CMT4H	150	6	883	6	**826**	**6.46%**
CMT5H	199	9	1044	9	**997**	**4.50%**
CMT6H	50	6	557	6	**556**	**0.18%**
CMT7H	75	11	943	11	**901**	**4.45%**
CMT8H	100	9	899	9	**869**	**3.34%**
CMT9H	150	14	1207	14	**1186**	**1.74%**
CMT10H	199	19	1499	**18**	**1441**	**3.87%**
CMT11H	120	4	830	4	**824**	**0.72%**
CMT12H	100	5	635	5	**628**	**1.10%**
CMT13H	120	11	1565	11	**1556**	**0.58%**
CMT14H	100	10	824	10	**821**	**0.36%**
			Average Improve Ratio			**2.11%**

In most of the benchmark instances (CMTnT), the performance of the AMSPSO can produce a better results compared with the conventional PSO algorithm, even if the customer size is increased from 50 to 199. In the instance of CMT2T and CMT3T, the AMSPSO can respectively reduce the total cost by 1.98% and 2.54% within 500 iteration times which is 50% of the conventional approach. The average improve ratio of the CMTnT is about 0.92%. The simulation results prove that the proposed AMSPSO can realize the less cost than conventional PSO algorithm [2]. In addition, the AMSPSO can further reduce one vehicle usage for CMT7T. The conventional PSO algorithm needs 1,000 iteration times to achieve the same level results. In the other two instance sets (CMTnQ and CMTnH), the proposed AMSPSO can achieve better performance than the conventional PSO algorithm, even if pickup ratios are increased to 25% and 50%, as shown in Table 5 and 6. The average improve ratio can reach 1.33% and 2.11%, respectively. The vehicle usage of CMT7Q and CMT10H is also reduced by AMSPSO. The maximum improve ratio is 4.31% and 6.46% for CMTnQ and CMTnH, respectively. A simulation result of the particle number changes is illustrated on Figure 3. The QPSO (*GroupC*) is taken about the half of the total particle numbers, which means the search performance is much better than PSOPC (*GroupB*) and PSO-RTVIWAC (*GroupA*). However, the particle number of QPSO is decreased after 250 iteration times. It proved that proposed punishment mechanism successfully avoid the QPSO to accumulate its credibility for taking more particles into its own group. In the end of the iteration times, PSOPC and PSO-RTVIWAC performs well for searching a better solution on middle area and local area. This simulation result proves that the proposed adaptive multi-swarm strategy can let the multi-swarm group to compete with each other for producing a better performance.

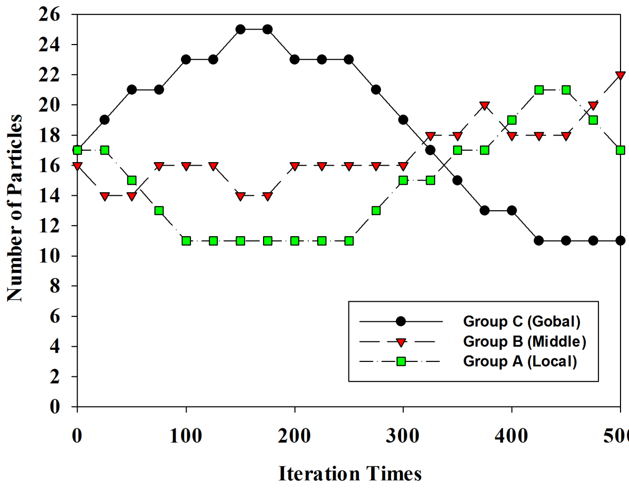

Figure 3. The simulation result of particle number changes

The above simulation results prove 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 AMSPSO.

5. Contributions to Existing Industrial Applications

The PSO algorithm is widely employed to deal with the optimization problem for industrial applications such as wireless sensor network [18], power system optimization [19], motor control [20], production scheduling [21] and so on. In the above, the PSO algorithm can produce a better performance than other conventional algorithms even if the problem space with high dimension. As a trade-off, the PSO algorithm is still time-consuming, since the PSO algorithm requires over several hundred iterative process to produce the better performance. The all of the above researches mentioned that once applying the PSO algorithm into real-time applications or real applications, the hardware implementation is highly required. However, it is very difficult to develop a generic PSO hardware to support various PSO applications, since the applications require different PSO algorithms to maintain its performance. Furthermore, the above mentioned applications require high adaptive capability for dynamic environment.

The proposed AMSPSO owns two features to meet the above requirements. The first feature is that the AMSPSO integrated nine different PSO into (15) to (21). The type of the PSO algorithm can be regulated by the particle motion coefficients. Compared with the conventional PSO algorithms, the hardware implement of AMSPSO can provide higher flexibility for various applications [22, 23]. In the [22, 23], the hardware implementation of AMSPSO can achieve twice processing speed compared with the hardware implementation of the conventional PSO algorithms. Another feature is the adaptive multi-swarm strategy which leads AMSPSO to provide very high adaptability for large-scale problem space or dynamic environment. AMSPSO exploits different PSO algorithms to cooperate with each others for preventing the particle swam from premature convergence as shown in section of simulation. The AMSPSO is expected to produce better performance for existing industrial applications based on the above two features.

6. Conclusions and Future Works

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 can reduce 50% iteration times of the conventional approach. The maximum improve ratios are 2.54%, 4.31% and 6.46% when the pickup ratios are 10%, 25% and 50%. In addition, the proposed adaptive multi-swarm strategy can let the multi-swarm groups to compete with each other for producing a better search performance. The AMSPSO can solve the CVRPPD with the less transportation cost. Furthermore, some new best known solutions of the benchmark problems are also found by the proposed AMSPSO. As the future work, the AMSPSO will compare with others similar approaches to further evaluate the performance under different kinds of VRP instances and real industrial applications. The parameter optimization of the AMSPSO will be carried out by different kinds of VRP instances.

Acknowledgement. This work was supported by Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research (No. 26280017) and Grant-in-Aid for Young Scientists (No. 15K21435). The authors thank Yang Zeng, Ying Deng, Qi Liang, and Yu Zhang for their efforts to do the paper work in a very restricted time frame during the paper preparation.

References

[1] Chen, K.-T., Jia, M., Zhang, J. and Baba, T. (2012) A flexible hardware architecture for particle swarm optimization. *J. of Signal Process.* 16(6): 519-526. doi:10.2299/jsp.16.519

[2] Ai, T.J. and Kachitvichyanukul, V. (2009) A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research* 36(5): 1693-1702. doi:10.1016/j.cor.2008.04.003

[3] Dantzig, G.B. and Ramser, J.H. (1959) The truck dispatching problem. *Management science* 6(1): 80-91. doi:10.1287/mnsc.6.1.80

[4] Kennedy, J. and Eberhart, R. (1995) Particle swarm optimization. In *Proceedings of IEEE Int. Conf. on Neural Networks* (Perth: IEEE), 1942-1948. doi:10.1109/ICNN.1995.488968

[5] Ai, T.J. and Kachitvichyanukul, V. (2009) Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem *Computers & Industrial Engineering* 56(1): 380-387. doi:10.1016/j.cie.2008.06.012

[6] Wu, B., Wang, W., Zhao, Y., Xu, X. and Yang, F. (2006) A novel real number encoding method of particle swarm optimization for vehicle routing problem. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation* (Dalian: IEEE), 3271-3275. doi:10.1109/WCICA.2006.1712972

[7] Liu, J., Sun, J. and Xu, W. (2006) Improving quantum-behaved particle swarm optimization by simulated annealing. In *Proceedings of Int. Conf. on Intelligent Computing* (Kunming: Springer), 130-136. doi:10.1007/11816102_14

[8] Zhu, H., Tanabe, Y. and Baba, T. (2008) A Random time-varying particle swarm optimization for

the real time location systems. *IEEJ Trans. on Electronics, Information and Systems* 128-C(12): 1747-1760. doi:10.1541/ieejeiss.128.1747

[9] He, S., Wu, Q.H., Wen, J.Y., Saunders, J.R. and Paton, R.C. (2004). A particle swarm optimizer with passive congregation. *Biosystems* 78(1): 135-147. doi:10.1016/j.biosystems.2004.08.003

[10] Bratton, D. and Kennedy, J. (2007) Defining a standard for particle swarm optimization. In *Proceedings of Swarm Intelligence Symposium* (Honolulu: IEEE), 120-127. doi:10.1109/SIS.2007.368035

[11] Shi, Y. and Eberhart, R.C. (1999) Empirical study of particle swarm optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation* (Washington: IEEE), 1945-1950. doi:10.1109/CEC.1999.785511

[12] Ratnaweera, A., Halgamuge, S. and Watson, H.C. (2004) Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. on Evolutionary Computation* 8(3): 240-255. doi:10.1109/TEVC.2004.826071

[13] Eberhart, R.C. and Shi, Y. (2001) Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the 2001 Congress on Evolutionary Computation* (Seoul: IEEE), 94-100. doi:10.1109/CEC.2001.934376

[14] Krohling, R.A. (2004) Gaussian swarm: a novel particle swarm optimization algorithm. In *Proceedings of IEEE Conference on Cybernetics and Intelligent Systems* (Singapore: IEEE), 372-376. doi:10.1109/ICCIS.2004.1460443

[15] Salhi, S. and Nagy, G. (1999) A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *J. of the Operational Research Society* 50(10): 1034-1042.

[16] Chen, K.-T., Dai, Y., Fan, K. and Baba, T. (2015) Performance Analysis of AMSPSO on Capacitated Vehicle Routing Problem with Time Window. In *Proceedings of RISP Int. Workshop on Nonlinear Circuits and Signal Processing* (Kuala Lumpur:RISP), 334-337.

[17] Chen, K.-T., Dai, Y., Fan, K. and Baba, T. (2015) A Particle Swarm Optimization with Adaptive Multi-Swarm Strategy for Capacitated Vehicle Routing Problem. In *Proceedings of the first Int. Conf. on Industrial Networks and Intelligent Systems* (Tokyo:EAI), (pp. to be appear).

[18] Kulkarni, R.V. and Venayagamoorthy, G.K. (2011) Particle Swarm Optimization in Wireless-Sensor Networks: A Brief Survey. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 41(2): 262-267. doi:10.1109/TSMCC.2010.2054080

[19] del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.-C. and Harley, R.G. (2008) Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. *IEEE Trans. on Evolutionary Computation* 12(2): 171-195. doi:10.1109/TEVC.2007.896686

[20] Mu, S., Tanaka, K. and Nakashima, S. (2013) Intelligent Control of Ultrasonic Motor Using PSO Type Neural Network. In *Proceeding of 14th ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (Honolulu: IEEE), 605-610. doi:10.1109/SNPD.2013.66

[21] Tang, L. and Wang, X. (2010) An Improved Particle Swarm Optimization Algorithm for the Hybrid Flowshop Scheduling to Minimize Total Weighted Completion Time in Process Industry. *IEEE Trans. on Control Systems Technology* 18(6): 1303-1314. doi:10.1109/TCST.2009.2036718

[22] Fan, K., Chen, K.-T, Wang, P. and Baba, T. (2015) Two-level Pipeline Structure of Particle Swarm Optimization. In *Proceedings of RISP Int. Workshop on Nonlinear Circuits and Signal Processing* (Kuala Lumpur:RISP), 178-181.

[23] Wang, P., Chen, K.-T., Fan, K. and Baba, T. (2015) A Hardware Implementation of Particle Swarm Optimization with Adaptive Multi-Swarm Strategy. In *Proceedings of RISP Int. Workshop on Nonlinear Circuits and Signal Processing* (Kuala Lumpur:RISP), 182-185.