Routing of Logistics Distribution Vehicles using Cloud Adaptive Mean Particle Swarm Optimization

Longlong Liu

Economics and Management Department, Shangluo University,
Shanxi 726000, China.

Abstract **— Due to its fixed inertia weight, the standard Particle Swarm Optimization (PSO) algorithm is easy to fall into local optimal solution when used to solve vehicle routing problems. To overcome this problem we propose a cloud adaptive adjustment strategy for inertia weights. The particles are divided into three groups according to the fitness of each particle, and different inertia parameter generating methods are used. The inertia weights in the general group is adaptively adjusted by X-conditional cloud generator. The cloud droplet has bias stability and randomness property, this not only preserves the swarm diversity, but also improves the convergence speed of the algorithm. Concurrently, the notion of mean is introduced into the PSO algorithm, and the velocity updating formula of the particle is modified. Finally, a cloud adaptive mean particle swarm optimization (CAMPSO) algorithm is proposed for the vehicle routing problem. The simulation experiments show that the proposed algorithm is greatly better than standard PSO and adaptive PSO in terms of global optimization ability and convergence speed.**

Keywords - logistics distribution; vehicle routing problem; particle swarm optimization; cloud model.

I. INTRODUCTION

With the development of the market, the logistics plays a more and more important role in an enterprise's economy growth and becomes its "third profit source". Logistics distribution is the key link of the logistics activities, and the driving paths of the distribution vehicles directly affect the efficiency, cost and service .(4VRP) [1].

In recent years, scholars have put forward a lot of algorithms to solve VRP, mainly including precise algorithms and approximate algorithms [2]. The precise algorithms, such as branch and bound, linear programming and so on, can obtain the global optimal solution to a problem. But the calculation of them will increase exponentially as problem size increases, so they are only suitable for solving small problems. The approximate algorithms, such as, genetic algorithm (GA), tabu search (TS) algorithm, particle swarm optimization (PSO) algorithm and so on, can only get approximate optimal solutions to the problems, and they are the focus of scholars. Because of its simple concept, easy implementation and quick convergence, PSO algorithm is widely used [3]-[4]. However, the standard PSO algorithm often falls into local optimal solution due to the fixed inertia parameter. In order to avoid this shortcoming, an adaptive particle swarm optimization (APSO) algorithm is proposed in [5], whose inertia weight is decreased linearly. The experiments indicate that APSO algorithm improves the global search ability of PSO algorithm, but it can not actually reflect the actual optimization search process, and needs to be further improved.

Based on above analysis, we propose a cloud adaptive mean particle swarm optimization (CAMPSO) algorithm, which combining the theory of cloud model. The population is divided into three groups according to the fitness of every individual, and different inertia weight generating strategies are adopted in different groups. For the better group, a smaller inertia weight is selected to speed up the convergence speed of CAMPSO algorithm. For the common group, the inertia parameter is adjusted dynamically by the X-conditional generator of the cloud model. Depending on the stable tendency and randomness of cloud droplet, the population diversity and convergence speed of the algorithm are improved. For the worse group, a higher inertia weight is got. At the same time, the notion of mean is introduced and the "cognitive" and "social" sections of particles' velocity updating formula are modified. Finally, the performance of CAPSO algorithm is verified by random examples and standard examples.

II. PROBLEM FORMULATION

The VRP is defined on a complete undirected network with a node set $K = \{0,1,\dots,k\}$ including one depot (node 0) and k customers. m same vehicles are required for the distribution and the capacity of every vehicle is q. Each customer i $(i = 1,2,\cdots,k)$ has a known request $g_i < q$. The distance between customer i and customer j is d_{ij}. Then the VRP can be formulated as follows.

$$\min \quad \mathbf{Z} = \sum_{i=0}^{K}\sum_{j=0}^{K}\sum_{s=1}^{m} d_{ij}x_{ijs} \tag{1}$$

$$\text{ST:} \quad \sum_{i=1}^{K} g_i y_{is} \le q, \quad s = 1,2,\cdots,m; \tag{2}$$

$$\sum_{s=1}^{m} y_{is} = 1, \quad i = 1,2,\cdots K; \tag{3}$$

$$\sum_{i=1}^{K} x_{ijs} = y_{js}, \quad j = 1,2,\cdots K, s = 1,2,\cdots,m; \tag{4}$$

$$\sum_{j=1}^{K} x_{ijs} = y_{is}, \quad i = 1,2,\cdots K, s = 1,2,\cdots,m; \tag{5}$$

$$x_{ijs} = 0 \ or \ 1; \tag{6}$$

$$y_{is} = 0 \ \ or \ \ 1. \tag{7}$$

The 0-1 variables x_{ijs} (6) are equal to 1 if and only if vehicle s visits customer j from customer i. The 0-1 binary variables y_{is} (7) take value 1 if and only if customer i is served by vehicle s. The objective function (1) represents the total travelling distances, to be minimized. Eq. (2) is the capacity constraint. Eq. (3) defines that every customer can be visited by only one vehicle. Eqs. (4) and (5) represents the relationship between x_{ijs} and y_{is}.

III. PSO ALGORITHM

The two key steps of PSO algorithm are: 1) generating an initial population of random solutions; 2) updating each particle's velocity and position at every iteration according to its personal best position and the global best particle found so far. The position of the i th particle in the d -dimensional search space at iteration t can be represented as $x_i^t(x_{i,1}^t, x_{i,2}^t, \cdots, x_{i,d}^t)$ and its velocity can be denoted by $v_i^t(v_{i,1}^t, v_{i,2}^t, \cdots, v_{i,d}^t)$. Each individual owns its best position $p_i^t = (p_{i,1}^t, p_{i,2}^t, \cdots, p_{i,d}^t)$ corresponding to the personal best objective value got so far at iteration t. The global best particle $p_g^t = (p_{g,1}^t, p_{g,2}^t, \cdots, p_{g,d}^t)$ is the best particle found so far in the whole population. The velocity and position of each particle are updated using Eq. (8) and (9):

$$v_{i,j}^{t+1} = wv_{i,j}^t + c_1 r_1 (p_{i,j}^t - x_{i,j}^t) + c_2 r_2 (p_{g,j}^t - x_{i,j}^t) \tag{8}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \qquad j = 1,2,\cdots,d \tag{9}$$

where w is named inertia weight; c_1 and c_2 are learning parameters, and they are usually equal to 2; r_1 and r_2 are independent random numbers generated uniformly between 0 and 1; $c_1 r_1 (p_{i,j}^t - x_{i,j}^t)$ is called "cognitive" section, and $c_2 r_2 (p_{g,j}^t - x_{i,j}^t)$ is named "social" section.

IV CAMPSO ALGORITHM FOR VRP

A. Cloud Theory

Cloud theory is developed from the concept of "membership cloud and language atom model" and is now widely used in the fields of intelligent control and data mining [6]. The cloud model is characterized by expected (Ex), entropy (En) and hyper entropy (He). Ex is the expectation value in the space of the discourse domain. En is used to expresses the uncertainty of the qualitative concept and it depends on the randomness and fuzziness of the droplets. He is the entropy of the En. It is used to measure the uncertainty of En. The generation process of cloud droplet is called cloud generator, including standard normal cloud generator and X-conditional cloud generator.

B. Cloud Adaptive Adjustment of Inertia Weight

The research in [7] shows that a larger w value can help PSO algorithm jump out of local optimal solution and search global optimal solution. While a smaller w is useful for searching local optimization and accelerating the convergence speed. So in order to get better solution, w can be set to 0.9 at the initial steps of PSO algorithm and is equal to 0.4 in the later stage. According to paper [7], combining the cloud model theory, we propose a cloud adaptive adjustment method for inertia weight.

Suppose f_i to represent the fitness value of the i th particle at k iteration, we can obtain the average fitness f_{avg} of the population according to the formula $f_{avg} = \frac{1}{N}\sum_{i=1}^{N} f_i$, where N indicates the particle swarm size. f_{avg}' and f_{avg}'' are the average fitness of the individuals, whose fitness are less than and greater than f_{avg}. We divide the particles into three groups by f_{avg}' and f_{avg}'', and adopt different inertia parameter generating strategies in different group.

(1) $f_i < f_{avg}'$. These particles that satisfy above condition are outstanding individuals in the population. They are already close to the global optimal solution. In order to accelerate the convergence speed of the algorithm, smaller inertia parameter should be taken. $w = 0.2$ in this paper.

(2) $f_{avg}' < f_i < f_{avg}''$. The inertia parameter of general particles, whose fitness values are between f_{avg}' and f_{avg}'', will be adjusted by the X-conditional generator of the cloud model. The adjusting method is as follows:

Step1: $Ex = f_{avg}'$

Step2:

$En = (f_{avg}{}' - f_{\min}) / C_1$ // C_1 *is the* control parameter

Step3:

$He = En / C_2$ // C_2 *is the* control parameter

Step4: $En' = normrnd(En, He)$

Step5: $w = 0.9 - 0.5 \times e^{-(f_i - Ex)^2 / 2(En')2}$

With the evolution of the algorithm, the fitness of particles will become smaller and smaller. And $0 < e^{-(f_i - Ex)^2 / 2(En')2} < 1$, thus, $w \in [0.4, 0.9]$.At the same time, w will become small as the particles' fitness become small. It ensures the excellent individuals own small w.

(3) $f_i > f_{avg}''$. These particles, whose fitness are higher than f_{avg}'', are the bad individuals in the population. We set $w = 0.9$.

C. Cloud Parameters Selection

The entropy affects the degree of the cloud model. According to "3En" rule, the coverage of the cloud model will increase with the increasing of entropy. Therefore, in order to improve the speed and precision of the algorithm, we set $C_1 = 6p(T+1)$ (p is the population size and T is the current iterative number). The hyper entropy determines the discrete degree of cloud model. The algorithm will lose stability if hyper entropy is too large and will lose randomness if it is too small. So a big hyper entropy value should be set at the initial stage in order to expand the search space and a small hyper entropy value should be given in the later stage to improve search precision. In this paper, we give $C_2 = 15 - (T - \frac{p}{2})^2$.

D. Encoding and Decoding

(1)Encoding method. The encoding strategy in [8] is adopted. The ideal of the encoding method is that a driving scheme of a vehicle scheduling problem with K clients is represented by a 2K-dimensional space vector. The 2K-dimensional vector of particle i is composed of two K-dimensional vectors: Z_{ix} and Z_{iy}. Z_{ix} expresses the delivery vehicles, and Z_{iy} indicates the vehicles' running orders in the path.

(2)Decoding method. First, determine the service vehicle of particle i.Take the integer of Z_{ix} and get all vehicles which are assigned to particle i.Secondly, give every vehicle's running order in the path. Look for all customers served by every vehicle (suppose the vehicle is j) and sort them from small to large according to their Z_{iy} value. It is the driving order of vehicle j.

E. Fitness Function

In order to process the algorithm and knock out the infeasible solutions in the particle population, we add the vehicle load constraint (2) to the objective function (1) and give a severe penalty to the infeasible particles. The fitness function is as follow:

$$\min \ \mathbf{Z} = \sum_{i=0}^{K}\sum_{j=0}^{K}\sum_{s=1}^{m} d_{ij}x_{ijs} + M(\sum_{s=1}^{m}\max(\sum_{i=1}^{K} g_i y_{is} - q, 0)) \quad (10)$$

where M is the penalty coefficient and is usually given a large positive value.

F. CAMPSO Algorithm for VRP

It is known that, the difference between individuals' and population's optimal values are obvious in the later iterations of standard PSO algorithm. It preserves the diversity of the swarm, but is not conducive to search the global optimal solution. Aiming at this shortcoming, we introduce the concept of mean in the algorithm. As shown in the formula (11), we modify the "cognitive" and "society" sections of the speed updating formula by linear combination of the individuals' and population's optimal values. On that basis, a cloud adaptive mean particle swarm optimization algorithm is proposed.

$$\begin{aligned} v_{i,j}(t+1) = wv_{i,j}(t) + c_1 r_1[\frac{p_{i,j} + p_{g,j}}{2} - x_{i,j}(t)] + \\ c_2 r_2[\frac{p_{i,j} - p_{g,j}}{2} - x_{i,j}(t)] \quad j = 1, 2, \cdots, d \end{aligned} \quad (11)$$

In summary, the procedure of CAMPSO algorithm is as follows:

Step 1: Initialize algorithm parameters, such as learning parameters, maximum iteration number, and population size.

Step 2: Randomly generate the initial particle swarm, including the vehicles and their order in the path.

Step 3: Decode the initial population and calculate the fitness value of each individual. Set each particle to its own historical optimal individual, and select the optimal particle as the historical optimal individual of the population.

Step 4: Update the inertia weight as the way shown in section 4.2.At the same time, update the position and velocity of each particle.

Step 5: Compare the updated particles with their own historical optimal individual and the historical optimal individual of population, and determine whether or not to update.

Step 6: If reach the maximum iteration number, output the historical optimal individual of the population. Otherwise, go back to Step 4.

Step 7: Decode the optimal particle of the population, it is the optimal vehicle scheduling scheme.

V. SIMULATION ANALYSIS AND CONCLUSION

A. *Random Example*

The customer network in [2] is selected as the random example. It is solved by CAMPSO algorithm and the results are given in Table 1. It can be known that the number of the vehicles needed for the distribution is 8, and the total travelling distance is 835.40km.

TABLE I THE OPTIMAL DISTRIBUTION SCHEME

NO.	travelling path	distance/km	NO.	travelling path	distance /km
1	0→8→22→13 →5→7→6→0	137.8	5	0→16→17→26 →0	101.5
2	0→1→29→12 →10→0	138.9	6	0→25→3→30 →0	129.6
3	0→4→18→9→ 15→27→0	115.0	7	0→11→20→24 →19→21→0	111.2
4	0→28→2→23 →0	93.3	8	0→14→0	7.2

In order to evaluate the performance of CAMPSO algorithm, standard PSO algorithm and APSO algorithm [5] are also used to solve this example and their results are compared with CAMPSO algorithm in Figure.1.

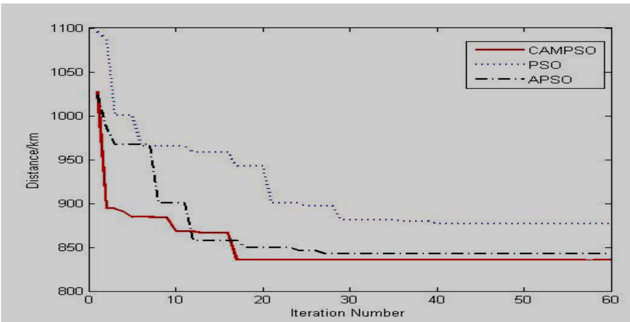

Figure.1 Convergence of algorithm

Figure.1 illustrates that, the convergence speed of APSO algorithm is faster than that of PSO algorithm, and CAMPSO algorithm is the fastest. It is easy to see CAMPSO algorithm has the strongest global search ability.

Then the robustness of CAMPSO algorithm is tested. The three algorithms are randomly run 20 times, and the indicators, such as, the best values (BV), the worst values (WV) and the average values (AV) are presented in table 2.

TABLE II COMPARISON ANALYSIS OF SIMULATION RESULTS

algorithm	Simulation results					
	BV/km	WV/km	AV/km	running time /s	successful searching rate[a] /%	average iteration number
PSO	835.40	1656.50	898.83	14.5	25	43.20
APSO	835.40	1342.13	860.76	13.6	40	34.73
CAMPSO	835.40	968.89	847.55	6.9	70	30.44

a : successful searching rate = (total running time – time found the best value)/ total running time x 100%

From table 2, we can see that the successful searching rates of these algorithms from big to small are CAMPSO algorithm, APSO algorithm and PSO algorithm, but the order of WV and AV is reverse. This phenomenon proves that the global search ability of CAMPSO algorithm is the strongest and APSO algorithm's and PSO algorithm's are weaker. At the same time, CAMPSO algorithm can get the best value in the least iteration number and time. It indicates that CAMPSO algorithm ensures the convergence speed while it improves the global search ability.

B. *Standard Examples*

In this section, the efficiency and stability of CAMPSO algorithm are analyzed. All these standard examples can be downloaded from http:// people. brunel. ac. uk/ ~mastjjb /jeb /info.html. They are solved by PSO algorithm, APSO algorithm and CAMPSO algorithm. The results are compared in Table 3.BKV is the best known value to the instance.

TABLE III THE RESULTS OF THE STANDARD EXAMPLE

instances	BKV/km	PSO			APSO			CAMPSO		
		BV/km	AV/km	error[b]/%	BV/km	AV/km	error /%	BV/km	AV/km	error /%
M-n33-k5	662	667	675	0.76	665	673	0.45	662	664	0.00
M-n46-k6	871	883	896	1.38	878	884	0.80	874	876	0.34
M-n35-k7	936	958	960	2.35	949	959	1.37	940	945	0.43
M-n65-k7	1021	1052	1067	3.04	1041	1062	1.92	1027	1039	0.59
M-n21-k9	1208	1224	1236	1.32	1219	1225	0.90	1213	1216	0.41

According to Table 3, these algorithms can't find all the BKVs of the instances, but the solutions got by CAMPSO algorithm are better than these got by PSO algorithm and APSO algorithm. For the small instances, such as M-n33-k5, the three algorithms can achieve good results. As the problem size increases, the stability of PSO algorithm and APSO algorithm is gradually reducing, but CAMPSO algorithm is still able to achieve better results.

Based on above observation, it is summarized that CAMPSO algorithm has stronger global search ability and faster convergence speed than other intelligent algorithms.

ACKNOWLEDGEMENTS

Fund Project:Shaanxi Provincial Education Department special research projects(14JK1214)

REFERENCES

[1] WU Jieming. "Vehicle routing optimization problem of logistics distribution".Computer Simulation,vol. 07, pp. 357-360, 2011.

[2] XU Maozeng, YU Guoyin, ZHOU Xiang, et al. "Low-carbon vehicle scheduling problem and algorithm with minimum-comprehensive-cost". Computer Integrated Manufacturing Systems, vol.21, No.7, pp.1906-1914,2015.

[3] ZHENG Jianru, ZHANG Guoli. "Based on improved particle swarm algorithm in mining vehicle delivery path". Coal Technology, vol.01,pp.207-208, 2013.

[4] WEI Ming, JIN Wenzhou. "Discrete particle swarm optimization algorithm for vehicle routing problems". Computer Science, vol.04,pp.187-191, 2010.

[5] LUO Dongsheng, LIU Yanmin. "Adaptive PSO based on swarm diversity for VRPSPD". Computer Engineering & Science,vol. 07,pp.160-165, 2012.

[6] LI Deiyi, MENG Haijun, "SHI Xuemei. Membership clouds and membership cloud generators". Journal of Computer Research and Development, vol.06,pp.15-20, 1995.

[7] Shi Y, Eberhart R C. "A modified particle swarm optimizer, Proceedings of the IEEE Conference on Evolutionary Computation". Piscataway, NJ: IEEE Press,pp.69-73, 1998.

[8] WU Yuechun. "Study on optimization of logistics distribution route based on AMPSO". Journal of Lanzhou Jiaotong University, vol.01,pp.114-117, 2012.