I. INTRODUCTION

Routing Problem Vehicle (VRP) is a typical NP-hard problem and proposed first by Dantzig and Ramser in 1959. Recent years, it has been a hot research field in computer science, operations research and combination optimization. Many problems in our daily life can be abstracted as vehicle routing problem, such as logistics distribution, power dispatch, postal delivery, school bus and routing problem, etc. This problem is full of important theoretical significance and engineering value on improving production efficiency and improving economic efficiency. Those years, many scholars have tried to introduce the general heuristic algorithm, genetic algorithm, ant colony algorithm for VRP problems and have achieved some good results [1-3].

Particle swarm optimization (PSO) is a global optimization evolutionary algorithm, which was put forward by Kennedy and Eberhart first in 1995 to solve the optimization problem of continuous domain function [4-5]. PSO is affected by the history optimum and global optimum of the particles, which can quickly converge to the global optimum or the local optimum. Since PSO has the characteristics of easy implementation, simple structure and strong robustness, many scholars have used it to solve the problem of discrete domain in recent years. For example, shop scheduling problems, etc. Based on the quantum particle swarm algorithm, crossover and mutation operation are added in [6] to improve the local search ability of the algorithm. In [7], a two-way vehicle scheduling problem model is established with the basis of the particle swarm algorithm and the mountain climbing operation is introduced, which effectively solves the problem of logistics distribution; In paper [8], a new particle swarm optimization algorithm is designed, which introduces the local neighbor mechanism and can optimize infeasible solutions. The algorithm obtains comparatively satisfactory results in solving the vehicle routing problem with time window. Paper [9] introduces multigroup parallel way and different initial methods are applied for each subgroup, besides the opposite population poor particle will be replaced by memory particle to solve the vehicle routing problem with time windows.

In this paper, based on the characteristics of VRP, we put forward an improved particle swarm optimization algorithm VPSO. In the application process is mainly manifested in the coding mode, infeasible solution processing, update the position of the particle, mutation and so on. Experiments show that, VPSO can faster and better to find the optimal solution of the vehicle routing problem.

II. VEHICLE SCHEDULING PROBLEM DESCRIPTION AND MATHEMATICAL MODEL

This paper investigates Capacity Vehicle Routing Problem for short vehicle routing problem, which is the most basic problem in all vehicle scheduling problems. Specifically described as follows:

There are a central warehouse with No. 0 and k vehicles in total with the capacity \( q(i = 1, 2, \cdots, k) \); Moreover, \( N \) customer point transportation tasks needs to be completed with No.1 to N with customer demand \( g_1, g_2, \cdots, g_N \).
Each vehicle from the central warehouse deliver to each customer and finally back to the central warehouse; $c_{ij}$ represents the distance between the customer $i$ and customer $j$. Moreover, the mathematical model in [2] is introduced in this paper.

First, define the 0-1 variable:

$$ x_{ijk} = \begin{cases} 
1 & \text{if vehicle } k \text{ goes through the } i \text{-th customer point to the } j \text{-th one} \\
0 & \text{otherwise}
\end{cases} $$

$$ y_{ki} = \begin{cases} 
1 & \text{if the } i \text{-th custom's task is done by the } k \text{-th vehicle.} \\
0 & \text{otherwise}
\end{cases} $$

The mathematical model is established as follows:

$$\min Z = \sum_{k=1}^{K} \sum_{i=0}^{N} \sum_{j=0}^{N} c_{ij} x_{ijk} $$

$$\sum_{i=0}^{N} g_{y_{ij}} \leq q_{i}, \forall k $$

$$\sum_{i=1}^{N} y_{ki} = 1, \forall i $$

$$\sum_{i=1}^{N} x_{ijk} = y_{ij}, \forall j, k $$

$$\sum_{j=1}^{N} x_{ijk} = y_{kj}, \forall i, k $$

The constraints in the model ensure that the carrying capacity of each vehicle is not over the load, and each customer will receive the service until the last service is finished. Therefore, vehicle routing problem is just the minimum path in these conditions.

III. IMPROVED PARTICLE SWARM OPTIMIZATION (VPSO)

The standard particle swarm optimization algorithm is an optimization algorithm based on a population, in which the individual is called particle and each particle's trajectory is determined by the Global best position (Gbest) and the particles' historical optimal solution (Pbest).

Let particle $X_i(t) = (x_{i1}, x_{i2}, \cdots, x_{ip})$ for the $t$ generation in a $D$ dimensional space. The velocity is $V_i(t) = (v_{i1}, v_{i2}, \cdots, v_{ip})$, and the load, and each particle is used to update two parameters: global best position $G_{\text{best}} = (P_{g1}, P_{g2}, \cdots, P_{gd})$ and Particle’s past best position $P_{\text{best}} = (P_{p1}, P_{p2}, \cdots, P_{pd})$. In the iterative process, the position $X_i(t)$ is updated by the velocity $V_i(t)$ with the next iteration via the following equation:

$$X_i(t+1) = X_i(t) + V_i(t+1)$$

while velocity $V_i(t)$ can be calculated as below:

$$V_i(t+1) = w \times V_i(t) + c_1 \times \text{rand}(0,1) \times (P_{g} - X_i(t)) + c_2 \times \text{rand}(0,1) \times (G_{\text{best}} - X_i(t))$$

The learning constant $c_1$ and $c_2$ are popularly equal to 2. The inertia parameter $w$ is an important parameter, which affects the performance of the algorithm. To make the algorithm better convergence to global optimal solution, we generally set $w = 0.9$ initially and gradually decrease to 0.4 in a linear way with the increase of iteration $t$, moreover, it follows the equation as blow:

$$w = 0.9 - 0.5 \times \frac{t}{T}$$

Where $T$ is the maximum iteration number.

From the particle’s position change formula (2) and (3), the update information of each particle is derived from itself and the whole group. The particle can rapidly move to the global optimum and the local optimal in the iterative process. But in the late stage, the group diversity is reduced and particles are easy to fall into local optimum. In order to improve the particle swarm optimization algorithm, this paper, a random individual is added in the velocity update formula (3) to help particle $X_i(t)$ jump out of local optimum. Specific formula is:

$$V_i(t+1) = w \times V_i(t) + c_1 \times \text{rand}(0,1) \times (P_{g} - X_i(t)) + c_2 \times \text{rand}(0,1) \times (P_{p} - X_i(t)) + c_3 \times \text{rand}(0,1) \times (X_i(t) - X_j(t))$$

Where $c_3$ is an accelerating parameter, $X_j(t)$ is a randomly selected particles and satisfied $i \neq j$. The velocity of each particle is determined by the global best position, particle’s past best position and the random individual position. Weight coefficients $c_1, c_2, c_3$ are used to determine their importance.
IV. VPSO FOR VEHICLE ROUTING PROBLEM

A. Particles of encoding and decoding

Using the particle swarm algorithm to solve practical problems, what we need to do first is the particle encoding. In this paper, we use the integer encoding method in [6].

Step1. The central warehouse with No.0, N customer with No.1 to N. Each particle is represented by a N dimensional integer vector \( X = [a_1, a_2, \ldots, a_N] \), where \( a_i \) is an integer from 1 to N, corresponding to the \( a_i \)-th customer.

Step2. Each particle according to the customer loading is not more than the vehicle capacity principle, and ultimately in accordance with the vehicle arrangement of customers to get the solution vector. If \( g_{a_1} + g_{a_2} + \cdots + g_{a_{i-1}} \leq q_i \) and \( g_{a_1} + g_{a_2} + \cdots + g_{a_{i-1}} + g_{a_i} > q_i \), then 1-th vehicle’s customer order is \([a_1, a_2, \ldots, a_{i-1}]\); similarly if \( g_{a_{i+1}} + g_{a_{i+2}} + \cdots + g_{a_{N-1}} \leq q_i \) and \( g_{a_{i+1}} + g_{a_{i+2}} + \cdots + g_{a_{N-1}} + g_{a_N} > q_i \), the 2-th vehicle’s customer order is \([a_{i+1}, a_{i+2}, \ldots, a_N]\). In this way, when vehicle passes all the customer point, the customer order is obtained and corresponding solution vector is \( M = [b_1, b_2, \ldots, b_k] \).

For example, there are 2 vehicles in total with the capacity of 8, 8 customer points with demand \([1,2,1,4,1,4,2,2] \). If the position vector of a particle is \( X = [1,3,4,5,2,7,6,8] \), according to the above method, the corresponding distribution plan is:

Path 1: 0 → 1 → 3 → 4 → 5 → 2 → 0;
path2: 0 → 7 → 6 → 8 → 0;
solution vector: \( M = [5,8] \)

B. Initialization of particles

Initial particle generation process is:

Step1. Generates a particle randomly and calculates its solution vector;
Step2. If the final value of the solution vector \( bk \) is N, then the particle is a feasible solution. Otherwise adjusts the infeasible solution in accordance with the 4.4 and the particle will be discarded if the result of adjustment is still not feasible.

Step3. Repeat step 1 and step2, generate NP effective particles in total, and construct the initial group.

C. Standardized processing of particles

The particles are iterated by the formula (2) (3) (5), the component of the particle will appear decimal number and negative number. In order to ensure that every vector of each particle has a corresponding path arrangement, the particle is required to be standardized. Specific method is follows:

Step1. The value at each dimension of particle \( X \) can be obtained by the formula (2) if it is limited in a range of \((-k-1) \) to \(k-1 \), or replace it with boundary value directly.

Step2. By the above step, we can obtain \( X = [a_1, a_2, \ldots, a_N] \). Now, we replace \( a_i(j = 1,2,\ldots,N) \) with its ascending ordinal number \( c_i(j = 1,2,\ldots,N) \) to get a new \( X = [c_1, c_2,\ldots,c_N] \).

For example, given particles \( X = [-2,3,1,4,1,3,5,7,6,8] \), through standardized process, we obtain \( X = [1,2,4,3,5,7,6,8] \).

D. Adjustment of infeasible solution

Whether in the initial population or the algorithm in the iterative process, particles \( X \) may appear a lot of infeasible solution path (if the final value of the solution vector \( bk \) is not N). In order to guarantee the validity of the algorithm, this paper proposed a kind of adjustment strategy, which can adjust most of the particles to the feasible solution. Specific methods as follows:

Step1. Find out all the path of overloaded vehicle.
Step2. For each overloaded vehicle, in accordance with the load from small to large, the order of the customer to adjust out of the vehicle until the load is lower than the capacity.
Step3. When all the vehicle load is not more than the capacity, all the adjusted customers sort as \( s_1, s_2, \ldots, s_m \) with the order of corresponding customer demand \( g_s(k = 1,2,\ldots,m) \) from large to the small; Calculate the residual load of each vehicle \( \Delta q_1(i = 1,2,\ldots,k) \) for the current infeasible solution and rank the sequence with the order from small to large: \( \Delta q_1, \Delta q_2, \ldots, \Delta q_k \).
Step4. Follow the order of $\Delta q_i$, $\Delta q_{i+1}$, ..., find out the vehicle which can able to load customer $s_i$, if $\Delta q_{i+1} < g_{s_i} < \Delta q_{i+2}$, in the original path of the vehicle $i_k$ to find the adjacent customers $p$ and $q$ that satisfied distance $p \rightarrow s_i \rightarrow q$ is shortest. Insert customer $s_i$ between $p$ and $q$ in vehicle $i_k$. Modify the value of $\Delta q_{i+1}$.

Step5. In accordance with step 4, Arrange customer $s_1, s_2, \ldots, s_m$ in turns.

Step6. If all the customers $s_1, s_2, \ldots, s_m$ is arranged, recalculate the corresponding solution vector according to the adjusted path; otherwise particle is judged infeasible and set the objective function as infinite.

E. Single Point Mutation

Mutation operation is to change some certain location and then form a new individual, which is beneficial to improve the local search ability of the algorithm. In this paper, the mutation operation is for the optimal solution of each generation, and the single point mutation is carried out among different sub paths.

Specific method is as follows: setting a mutation probability $P_m$, if $P_m \geq \text{rand}(0,1)$, generated two mutation positions from different paths of optimal solution and then exchange them. If the new solution is better than the original, we set the new solution to the best solution.

For example: particle $[3,1,2,6,5,7,8,4]$ and it’s solution vector is $[4,8]$, corresponding to the two path: $0 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 6 \rightarrow 0$ and $0 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 4 \rightarrow 0$. When the path is feasible, from fig1, we know that the new path is obviously better than the original path.

F. The implementation of the improved algorithm for VRP

Particle swarm algorithm is applied to continuous space, while the VRP is a discrete integer programming problem, so we need to modify the algorithm for the specific application. The specific process is as follows:

Step1. Particles initialization

(1) Within the initialization method in 4.2, particles about NP in total are randomly generated and divided into two overlapping adjacent subgroups. The number of overlapping particles is $cd$, and the number of particles in each subgroup is $\frac{NP + cd}{2}$.

(2) Calculate the initial value of each particle, the historical optimal solution (Pbest) and the global best position (Gbest).

Step2. Repeat the following steps until the maximum number of iterations.

(1) In each subgroup, every particle is updated by formula(2)(5), then we employ a standardize process for particle via the method in 4.3, and adjust unfeasible solution according to 4.4, calculate the fitness value.

(2) Replace overlap particles with optimal location in two groups.

(3) Find out the global optimal solution of the current generation, and then searched it by Single Point Mutation with 4.5.

(4) Calculate the Global best position (Gbest) and the particles’ historical optimal solution (Pbest).

Step3. Finally, global best position is taken as the final optimal path, and the corresponding path length is the optimal path length.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In the present work, to compare results conveniently, we use Matlab 7.0 to write the program of particle swarm optimization (PSO) and the improved algorithm (VPSO) to solve the vehicle routing problem with the computer operating system 1.2GHz, 988RAM, WinXP.

A. Experiment 1

We take the data of paper[9] in our experiment 1. The vehicle routing problem has a central warehouse, 2 vehicles in total with the capacity of 8, and 8 customer points with demand $[1,2,1,2,1,4,2,2]$ . The following table gives the distance and the demand of the customers. Now it is required to arrange a suitable driving route so that the total mileage of the vehicle route is minimized. This paper tells us that the optimal solution path length is 67.5 and the path is arranged as follows: Path 1: $0 \rightarrow 4 \rightarrow 7 \rightarrow 6 \rightarrow 0$; path2: $0 \rightarrow 1 \rightarrow 3 \rightarrow 5 \rightarrow 8 \rightarrow 2 \rightarrow 0$. 

International Journal of Emerging Technology and Advanced Engineering
### Table 1. The Distance Between Customers (KM) and Demand

<table>
<thead>
<tr>
<th>Cij</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>7.5</td>
<td>9</td>
<td>20</td>
<td>10</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>6.5</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>7.5</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6.5</td>
<td>0</td>
<td>7.5</td>
<td>10</td>
<td>10</td>
<td>7.5</td>
<td>7.5</td>
<td>7.5</td>
</tr>
<tr>
<td>3</td>
<td>7.5</td>
<td>4</td>
<td>7.5</td>
<td>0</td>
<td>10</td>
<td>5</td>
<td>9</td>
<td>9</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>0</td>
<td>10</td>
<td>7.5</td>
<td>7.5</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>10</td>
<td>0</td>
<td>7</td>
<td>9</td>
<td>7.5</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>7.5</td>
<td>7.5</td>
<td>9</td>
<td>7.5</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>11</td>
<td>7.5</td>
<td>9</td>
<td>7.5</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>10</td>
<td>7.5</td>
<td>15</td>
<td>10</td>
<td>7.5</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>demand</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Parameter setting of PSO: population size $NP = 40$, $w = 0.6$, learning constant $c_1 = c_2 = 1.5$, the maximum iteration number $T = 200$.

Parameter setting of VPSO: population size $NP = 40$, $w = 0.6$, learning constant $c_1 = c_2 = 1.5$, accelerating parameter $c_3 = 1.0$, overlapping particles’ number $cd = 2$, the number of particles in each subgroup is 22, mutation probability $P_m = 0.2$, the maximum iteration number $T = 200$.

PSO, VPSO runs each 20 times and compares with the results of the algorithm proposed by [9]. The test results are showed in table 2.

### Table 2. The Optimal Path Length (KM) of the Three Algorithms

<table>
<thead>
<tr>
<th>Running times</th>
<th>Algorithm of paper [9]</th>
<th>PSO</th>
<th>VPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>2</td>
<td>67.5</td>
<td>69.0</td>
<td>67.5</td>
</tr>
<tr>
<td>3</td>
<td>71.0</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>4</td>
<td>67.5</td>
<td>69.0</td>
<td>67.5</td>
</tr>
<tr>
<td>5</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>6</td>
<td>72.0</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>7</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>8</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>9</td>
<td>71.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>10</td>
<td>67.5</td>
<td>69.0</td>
<td>67.5</td>
</tr>
<tr>
<td>11</td>
<td>67.5</td>
<td>69.0</td>
<td>67.5</td>
</tr>
<tr>
<td>12</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>13</td>
<td>69.0</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>14</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>15</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>16</td>
<td>67.5</td>
<td>69.0</td>
<td>67.5</td>
</tr>
<tr>
<td>17</td>
<td>69.0</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>18</td>
<td>67.5</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>19</td>
<td>70.0</td>
<td>67.5</td>
<td>67.5</td>
</tr>
<tr>
<td>20</td>
<td>67.5</td>
<td>70.0</td>
<td>67.5</td>
</tr>
</tbody>
</table>

262
From table 2, the algorithm in the paper [9] has 6 times without finding the optimal solution, and PSO has not found the optimal solution about 6 times, while the VPSO can find the optimal solution in 100%. Table 3 gives a more comparison results of the VPSO and PSO, including the times of achieve, not achieve times, average value, average running time, Average iteration number of optimal path.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Achieve times</th>
<th>Not achieve times</th>
<th>Average value</th>
<th>Average iteration number of optimal path</th>
<th>Average running times</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>14</td>
<td>6</td>
<td>68.0</td>
<td>38.308</td>
<td>9.475</td>
</tr>
<tr>
<td>VPSO</td>
<td>20</td>
<td>0</td>
<td>67.5</td>
<td>19.523</td>
<td>10.417</td>
</tr>
</tbody>
</table>

Experimental result shows that the search success probability increased significantly. Because of the existence of subgroup exchanges, mutation operation and adjustment of infeasible solution. The calculation time becomes relatively longer, but not so much. Consequently, we can get that the improved algorithm VPSO has a higher search efficiency and better stability. It is an ideal method for the VRP problem with fewer number of customers.

B. Experiment 2

In order to verify the effectiveness of VPSO in the process of dealing with more customers, this paper uses the medium scale example P-n22-k8(Here we select data from http://branchandcut.org/). The vehicle routing problem has a central warehouse, 8 vehicles in total with the capacity of 3000, 22 customer points and the optimal solution path length is 603.

PSO, VPSO runs each 10 times and compared. Parameter value setting as follows: population size $NP = 160$, overlapping particles’ number $cd = 10$, the number of particles in each subgroup is 85, $w = 0.6$, learning constant $c_1 = c_2 = 1.5$, accelerating parameter $c_0 = 1.0$, mutation probability $P_m = 0.2$, the maximum iteration number $T = 2000$. Table 4 gives the best solution(best), average solution(avg), worst solution(worst), variance(std).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>best</th>
<th>avg</th>
<th>worst</th>
<th>std</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPSO</td>
<td>633.3616</td>
<td>648.0531</td>
<td>666.9142</td>
<td>25.82468</td>
</tr>
<tr>
<td>PSO</td>
<td>650.3929</td>
<td>668.1819</td>
<td>683.3928</td>
<td>28.02677</td>
</tr>
</tbody>
</table>

With the increase of the number of customers, the search of the optimal solution is more difficult, and the effect of PSO and VPSO will have a certain effect. But from table 4, it can be seen that the results of VPSO have obvious advantages. Therefore, the experiment proves that the improved algorithm VPSO is effective and feasible to solve the problem of VRP.

VI. CONCLUSION

In this paper, a particle swarm optimization algorithm with Gauss mutation (VPSO) and neighborhood search is designed to solve the vehicle routing problem. In the process of solving vehicle routing problem, VPSO uses the integer encoding divided into two subgroups respectively iteration and enhances the search ability of the group. Finally, the simulation experiments show that the proposed algorithm can get the optimal solution faster and better. However, when solving the VRP with a larger scale, the algorithm in this paper will be more difficult with the customers increasing and the searching ability of the algorithm still needs to be improved.

Acknowledgement

Project supported by the Natural Science Foundation of Sichuan Education (14ZA0127) and Doctor Foundation of China West Normal University (12B022).

REFERENCES


