American Journal of Software Engineering and Applications
2016; 5(1): 1-6
Published online December 25, 2015 (http://www.sciencepublishinggroup.com/j/ajsea)
doi: 10.11648/j.ajsea.20160501.11
ISSN: 2327-2473 (Print); ISSN: 2327-249X (Online)

Vehicle Routing Problem Based on Particle Swarm Optimization Algorithm with Gauss Mutation

Ting Xiang, Dazhi Pan, Haijie Pei

College of Mathematic and Information, China West Normal University, Nanchong, China

Email address:
1198715106@qq.com (Ting Xiang), pdzzj@126.com (Dazhi Pan)

To cite this article:
Ting Xiang, Dazhi Pan, Haijie Pei. Vehicle Routing Problem Based on Particle Swarm Optimization Algorithm with Gauss Mutation. *American Journal of Software Engineering and Applications.* Vol. 5, No. 1, 2016, pp. 1-6. doi: 10.11648/j.ajsea.20160501.11

Abstract: In order to solve the vehicle routing problem, this paper introduces the Gauss mutation, which is based on the common particle swarm algorithm, to constitute an improved particle swarm algorithm (NPSO). In the process of solving vehicle routing problem, the NPSO is encoded by integer and proposes a new way to adjust the infeasible solutions. The particles are divided into two overlapping subgroups, and join the two-two exchange neighborhood search to iterate. Finally, the simulation experiments show that the proposed algorithm can get the optimal solution faster and better, and it has a certain validity and practicability.

Keywords: Particle Swarm Optimization, Vehicle Routing Problem, Gauss Mutation, Neighborhood Search

1. 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 basic particle swarm algorithm, a new algorithm (NPSO) with the Gauss mutation is proposed. In the process of solving the vehicle routing problem, we introduce a series measures including the integer encoding (see [6]), how to deal with the infeasible solution and the neighborhood search so as to improve the ability of the particle to jump out of the unfeasible solution region and the local optimal position.

2. 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 $q_i(i=1,2,\cdots k)$; 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:

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

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

The mathematical model is established as follows:

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

$$s.t\begin{cases}\sum_{i=1}^{N}g_i y_{ki}\le q_i & \forall k\\ \sum_{k=1}^{K}y_{ki}=1 & \forall i\\ \sum_{i=1}^{N}x_{ijk}=y_{kj} & \forall j,k\\ \sum_{j=1}^{N}x_{ijk}=y_{kj} & \forall i,k\end{cases} \tag{1}$$

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.

3. Particle Swarm Optimization Algorithm with Gauss Mutation

The standard particle swarm optimization algorithm is an optimization algorithm based on 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 $X_i(t)=(x_{i1},x_{i2},\cdots x_{iD})$ for the t generation in a D dimensional space. The velocity is $V_i(t)=(v_{i1},v_{i2},\cdots,v_{iD})$, global best position $Gbest=(P_{g1},P_{g2},\cdots P_{gD})$, particle's past best position $Pbest_i=(P_{i1},P_{i2},\cdots P_{iD})$. 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:

$$X_i(t+1)=X_i(t)+V_i(t+1), \tag{2}$$

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

$$V_i(t+1)=w\times V_i(t)+c_1\times rand(0,1)(P_{i\,best}-X_i(t))+c_2\times rand(0,1)(G_{best}-X_i(t)). \tag{3}$$

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, furthermore, it follows the equation as blow:

$$w=0.9-0.5*\frac{t}{T}, \tag{4}$$

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, based on the original individual, introduces a Gauss perturbation term. Specific way is: set a mutation probability P_c and randomly select individuals about $NP\times P_c$ from the population with NP in total. Gauss mutation is applied for each individual in the j-th dimension under the mutation formula :

$$X_{ij}=X_{ij}+L\times N(0,1)\times X_{ij}, \tag{5}$$

where X_{ij} is the j-th dimension of particle X_i, j is a random integer, L is the parameter of Gauss mutation; $N(0,1)$ is a random vector of the Gauss distribution, which is subject to a mean of 0 and 1 of the variance.

There is something that we have to mention here. $L\times N(0,1)\times X_{ij}$ in formula (5) is Gauss random perturbation term, which makes full use of the known information of the current such that it does not increase the diversity of the group and has advantageous to jump out of the local optima to carry on the global search, but also improves the search speed.

4. Improved Algorithm for Vehicle Routing Problem

4.1. 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 $X=[a_1,a_{2,}\cdots a_N]$, 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 $g_{a1}+g_{a2}+\cdots+g_{a_{b1}}\le q_1$ and $g_{a1}+g_{a2}+\cdots+g_{a_{b1}}+g_{a_{b1+1}}>q_1$, then 1-th vehicle's customer order is $[a_1,a_2,\cdots a_{b1}]$; similarly if $g_{a_{b1+1}}+g_{a_{b1+2}}+\cdots+g_{a_{b2}}\le q_2$ and $g_{a_{b1+1}}+g_{a_{b1+2}}+\cdots+g_{a_{b2}}+g_{a_{b2+1}}>q_2$, the 2-th vehicle's customer order is $[a_{b1+1},a_{b1+2},\cdots a_{b2}]$. In this way, when vehicle passes all the customer point, the customer order is obtained and corresponding solutions vector is $M=[b1,b2,\cdots bk]$.

For example, there are 2 vehicles in total with the capacity of 8, 8 customer points with demand $[1,2,1,2,1,4,2,2]$. 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; path 2: 0→7→6→8→0; solutions vector: $M=[5,8]$

4.2. 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.

4.3. Standardized Processing of Particles

The particles are iteratived 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 $X=[a_1,a_2,\cdots a_N]$. Now, we replace $a_i(i=1,2,\cdots,N)$ with it's ascending ordinal number $c_i(i=1,2,\cdots,N)$ to get a new $X=[c_1,c_2,\cdots,c_N]$.

For example, given particles $X=[-2.3,1,4,1.3,5,7,6.4,8]$, through standardized process, we obtain $X=[1,2,4,3,5,7,6,8]$.

4.4. 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 are divided into three parts:

The first part.

Step1. Calculate the residual load of each vehicle $\Delta q_i(i=1,2,\cdots k)$ for the current infeasible solution and rank the sequence with the order from small to large: $\Delta q_{i1},\Delta q_{i2},\cdots,\Delta q_{ik}$, where ik is the number of vehicle.The non-scheduled customer point is arranged as $s_1,s_2,\cdots s_m$ with the order of corresponding customer demand $g_{s_k}(k=1,2,\cdots,m)$ from large to the small.

Step2. For any customer $s_j(j=1,2,\cdots,m)$, find out some vehicle from $\Delta q_{ij}(j=1,2,\cdots k)$ with the order of $\Delta q_{i1},\Delta q_{i2},\cdots,\Delta q_{ik}$ such that the vehicle can load customer. Take customer s_j for example, if $\Delta q_{i_{l-1}}<g_{s_j}\le\Delta q_{i_l}$, we put customer s_j into the vehicle i_l. In the i_l -th path to find the adjacent two customer points p and q such that the distance $p\to s_j\to q$ is the shortest and insert s_j between p and q , modifying Δq_{il} at the same time.

Step3. According to the step2, if all of the customers have been arranged, directly into the third part, otherwise enter the second part.

The second part

Step1. Since the conduction in this step are similar to the step 1 at first part, so we omit here.

Step2. If $g_{s_1}\le\Delta q_{i_1}$, put the customer s_1 into vehicle i_1 directly, otherwise we find out the the minimum customer x of the vehicle i_1 with corresponding customer demand g_x such that $g_x+\Delta q_{i_1}\ge g_{s_1}$. If $g_x<g_{s_1}$, place s_1 at the location of x , consequently x becomes a customer point that is not scheduled and update g_{s_1} with g_x .

Step3. If the adjustment in step2 does not work for customer s_1, then consider whether customer s_1 should be put the into the next vehicle, Of course, s_2 can be applied with the same way.

Step4. Repeat step2 and step3 until that we can not find out a vehicle to arrange s_1 or $g_x=0$.

Step5. Repeat the step1, 2, 3, 4 for $s_1,s_2,\cdots s_t$ respectively until all the non-scheduled customers are traversed.

The third part

Particle vector and the corresponding solution will be calculated after the above two parts. If the final value of the solution vector is still not N, then particle is judged infeasible and set the objective function as infinite.

4.5. Neighborhood Search

With the iterative process, the group is easy to fall into local optimum, but the global optimal solution is near the local optimum. In this paper, to ensure that the algorithm convergence to the global optimum more quickly, global optimal solution is searched by a two-two exchange neighborhood search per 20 generations.

The specific method is stated as follows.

Set a neighborhood search parameter *sd* first, which represents neighborhood search times. Then randomly we select the $i-th$ and $j-th$ location of $Gbest$ and exchange their position each time. After the adjustment in 4.4 to get the new particle $\overline{Gbest}$. If $\overline{Gbest}$ is better than the previous $Gbest$, then replace $Gbest$ with $\overline{Gbest}$.

4.6. 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, about NP particles 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) (3), randomly select $floor(\frac{NP+cd}{2}\cdot P_c)$ particles to participate in the Gauss mutation by formula (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) If current iteration number is a multiple of 20, then global optimal solution is searched by a two-two exchange neighborhood search 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.

5. 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(NPSO) to solve the vehicle routing problem with the computer operating system 3.3GHz, 8.00GB, Win7.

5.1. 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→4→7→6→0; path2: 0→1→3→5→8→2→0.

Table 1. *The distance between customers (km) and demand.*

C_{ij}	0	1	2	3	4	5	6	7	8
0	0	4	6	7.5	9	20	10	16	8
1	4	0	6.5	4	10	5	7.5	11	10
2	6	6.5	0	7.5	10	10	7.5	7.5	7.5
3	7.5	4	7.5	0	10	5	9	9	15
4	9	10	10	10	0	10	7.5	7.5	10
5	20	5	10	5	10	0	7	9	7.5
6	10	7.5	7.5	9	7.5	7	0	7	10
7	16	11	7.5	9	7.5	9	7	0	10
8	8	10	7.5	15	10	7.5	10	10	0
demand	1	2	1	2	1	4	2	2	

Parameter setting of PSO: population size $NP=40$, learning constant $c_1=c_2=1.5$, the maximum iteration number $T=50$.

Parameter setting of NPSO: population size $NP=40$, learning constant $c_1=c_2=1.5$, the maximum iteration number $T=50$, overlapping particles' number $cd=2$, the parameter of Gauss mutation $L=1.8$; mutation probability $P_c=0.8$, neighborhood search parameter $sd=2$. PSO, NPSO 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.*

Running times	Algorithm of paper [9]	PSO	NPSO	Running times	Algorithm of paper [9]	PSO	NPSO
1	67.5	67.5	67.5	11	67.5	67.5	67.5
2	67.5	70.5	70.0	12	67.5	70.0	67.5
3	71.0	70.0	67.5	13	69.0	70.0	67.5
4	67.5	70.0	67.5	14	67.5	67.5	69.0
5	67.5	67.5	67.5	15	67.5	70.5	67.5

Running times	Algorithm of paper [9]	PSO	NPSO	Running times	Algorithm of paper [9]	PSO	NPSO
6	72.0	67.5	67.5	16	67.5	67.5	67.5
7	67.5	71.0	67.5	17	69.0	67.5	67.5
8	67.5	70.0	67.5	18	67.5	67.5	67.5
9	71.5	67.5	67.5	19	70.0	70.0	67.5
10	67.5	67.5	67.5	20	67.5	70.0	70.0

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 10 times, while the NPSO has only 3 times that the global optimal solution is not reached.The probability of NPSO finding the optimal solution is 85%.Table 3 gives a more comparison results of the NPSO and PSO, including the achieve times, the not achieve times, average running time, the best value(best), average value(avg), worst value(worst).

Table 3. *Comparison of experimental results with NPSO and PSO.*

algorithm	Achieve times	not achieve times	Average running time(s)	Best value	Avg value	Worst value
PSO	10	10	0.0833	67.5	68.85	71.0
NPSO	17	3	0.1397	67.5	67.83	70.0

Experimental results shows that the search success probability of the algorithm in paper [9] is 70% and PSO only 50%, however NPSO can reach as high as 85% and the search success probability increased significantly. Because of the existence of subgroup exchanges, neighborhood search, Gauss mutation, not feasible solution, the calculation time becomes relatively longer, but not so much. Consequently, we can get that the improved algorithm NPSO has a higher search efficiency and better stability. It is an ideal method for the VRP problem with fewer number of customers.

5.2. Experiment 2

In order to verify the effectiveness of NPSO in the process of dealing with more customers, this paper uses NPSO and PSO to run 10 times for VRP with different scale (Here we select data fromfrom http://branchandcut.org/). For each instance corresponds to a different parameter, we set as follows, learning constant $c_1 = c_2 = 1.5$; the parameter of Gauss mutation $L = 1.8$; mutation probablity $P_c = 0.8$; population size NP about 5~8 times the number of customers; $cd \in [2,20]$ and increased with the particle number NP ; neighborhood search parameter $sd \in [\frac{N}{3}, \frac{N}{2}]$. The final results are compared in table 4.

Table 4. *PSO and NPSO results for each instance.*

instance	Theoretical optimal solution	Parameter setting	algorithm	Best value	Average value	Worst value	Running time
P-n16-k8. vrp	450	$NP = 100$; $cd = 4$; $sd = 6$; $T = 1000$	PSO	451.34	451.46	451.95	37.01
			NPSO	451.34	451.34	451.34	38.16
P-n22-k8. vrp	603	$NP = 160$; $cd = 6$; $sd = 8$; $T = 2000$	PSO	627.30	647.36	680.21	175.72
			NPSO	602.72	618.78	639.36	180.09
B-n31-k5. vrp	672	$NP = 160$; $cd = 6$; $sd = 10$; $T = 2000$;	PSO	795.18	834.21	881.82	26.27
			NPSO	724.23	794.17	823.59	31.03
P-n40-k5. vrp	458	$NP = 200$; $cd = 10$; $sd = 14$; $T = 3000$;	PSO	790.87	829.76	914.70	41.90
			NPSO	636.26	735.26	778.36	47.94

With the increase of the number of customers, the search of the optimal solution is more difficult, and the effect of PSO and NPSO will have a certain effect. But from table 4, it can be seen that the results of NPSO have obvious advantages. We have to admit that there are many operations in NPSO and takes much time, but they are always in an acceptable range. Therefore, the experiment proves that the improved algorithm NPSO is effective and feasible to solve the problem of VRP.

6. Conclusion

In this paper, a particle swarm optimization algorithm with Gauss mutation (NPSO) and neighborhood search is designed to solve the vehicle routing problem. In the process of solving vehicle routing problem, NPSO 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

[1] Jun Li, Yaohua Guo. Theory and method of logistics distribution vehicle scheduling [M]. Beijing: China material press, 2001.

[2] Laporte G. The vehicle routing problem: an overview of exact and approximation algorithms [J]. European Journal of operational Research, 1992, 5(9): 345 一 358.

[3] Hongchun HU, Yaohua WU, Li LIAO. Optimization and application of logistics distribution vehicle [J]. Journal of Shandong University (Engineering Science), 2007, 37(4): 104-107.

[4] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [A]. Proceedings of the International Symposium on Micro Machine and Human Science [C]. Piscataway, NJ, USA: IEEE, 1995: 39-43.

[5] Kennedy J, Eberhart R. Particle swarm optimization [A]. Proceedings of the IEEE International Conference on Neural Networks [C]. Piscataway, NJ, USA: IEEE, 1995: 1942-1948.

[6] Zhen Huang. Hybrid quantum Particle Swarm Optimization algorithm for vehicle routing problem [J]. Computer Engineering and Applications, 2013. 49(24): 219-223.

[7] Dongqing Ma, Wei Wang. Logistics distribution vehicle scheduling based on improved particle swarm optimization [J]. Computer Engineering and Applications, 2014, 50(11): 246-270.

[8] Yaohua Wu, Nianzhi Zhang. Modified Particle Swarm Optimization algorithm for vehicle routing problem with time windows [J]. Computer Engineering and Applications, 2010, 46(15): 230-234.

[9] Ya Li, Dan Li, Dong Wang. Improved chaos particle swarm optimization algorithm for vehicle routing problem [J]. Application Research of Computers, 2011, 28(11): 4107-4110.

[10] Ning Li, Tong Zou. Particle swarm optimization for vehicle routing problem [J]. Journal of Systems Engineering, 2004, 19(6): 596-600.

[11] Bing Wu. Research and application of particle swarm optimization algorithm for vehicle routing problem [D]. Zhejiang University of Technology, 2008.

[12] Yuanbing Mo, Fuyong Liu. Artificial glowworm swarm optimization algorithm with Gauss mutation [J]. Application Research of Computers, 2013, 30(1): 121-123.

[13] Xing Liu, Guoguang He. Study on tabu search algorithm for stochastic vehicle routing problem [J]. Application Research of Computers, 2007, 43(24): 179: 181.