Journal of Applied Science and Engineering Innovation, Vol.6, No.2, 2019, pp. 99-103
ISSN (Print): 2331-9062
ISSN (Online): 2331-9070

Improved Particle Swarm Optimization Based on Tabu Search for VRP

Harrison Yonghao Zeng[1]

[1] *Qingdao University, No.308 Ningxia Road, Qingdao, Shandong, China*

Abstract*:* Particle swarm optimization (PSO) is a swarm intelligence optimization algorithm which exhibits characteristics such as simplicity and fast convergence. However, when applying particle swarm optimization to solve the vehicle route problem (VRP), some issues often emerge. For example, the solution area is integer domain and the path cannot be effectively coded. To better tackle these issues, an improved particle swarm optimization algorithm based on tabu search is thus proposed to solve the VRP. Firstly, the forward direction of particle swarm optimization (PSO) algorithm is modified to transform the solving region of particles from real domain to integer domain, so that algorithms can be adapted to the optimization of integer domain space. Secondly, based on tabu search algorithm, path coding and algorithm design are carried out to solving the path planning problem. Finally, simulation experiments run on Solomon data sets can demonstrate that the improved PSO algorithm in this paper is superior to the contrast algorithm in terms of delivering more accurate results and higher convergence speed.

Keywords Particle Swarm Optimization; Tabu Search; Vehicle Routing Problem

INTRODUCTION

The Particle Swarm Optimization(PSO) algorithm, proposed by R. Eberhart and J. Kennedy in 1995[J. Kennedy, *et. al.*, 1997], is a method based on population stochastic optimization. The method aims to simulate the social behaviors of fish schooling or bird flocking. In other words, it randomly creates a swarm of particles, each representing a candidate solution to the problem. The position of every particle is evaluated by the target function and iteratively reaches the best solution. The Particle Swarm method has already been implemented on various types of optimization problems, including AI learning, production management, robot control, image processing. However, its application often encounters a premature convergence to a local best solution, thus resulting in an inability to effectively find the best solution. Many improvements for PSO are therefore developed to tackle this premature issue: Kennedy and Eberhart's Particle Swarm Optimization algorithm in a Discrete Space [Kennedy, *et. al.*, 1997], Clerc's particle Swarm Optimization algorithm with a shrinkage factor [M. Clerc, et. al., 2002], Tian Na et al.'s Particle Swarm Optimization algorithm with quantum behavior [Na Tian, *et. al.*, 2011], the Hybrid Optimization algorithm proposed by Ma Chao et al. integrates the genetic algorithm and PSO algorithm [Chao Ma, *et. al.*, 2011]. All of these above modifications have gained, to some degree, positive feedback.

The VRP (vehicle route problem) is a typical combination optimization problem presented by Dantzig and Ramser in 1959 [Dantzig, *et. al.*, 1959]. Hitherto, researchers have done lots of research on VRP with different conditions and proposed various solutions. These solutions can be categorized into exact and heuristic algorithms. Amongst them, novel heuristic algorithms, such as artificial neutral net, genetic algorithms, simulated annealing algorithms and tabu algorithms provide new insights and measures for solving complex VRPs. Currently, many researchers are trying to use PSO to solve VRP and have proposed several possible algorithm designs. Sen Guo and Guihe Qin use discrete domain PSO on VRP [Sen, *et. al.*, 2016], Wang Z and Guo J use PSO on Multiple-Objective Travelling Salesman Problems (MOTSP) [Wang, *et. al.*, 2015], Yi Li and Baichuan Lu employ chaotic PSO to VRP with time windows [Yi, *et. al.*, 2012], and have all reached desired results. However, when putting PSO into practical use, the discontinuity of the solution area and the various conditions of VRP leads to lots of invalid solutions in the problem-solving process. The algorithms turn out to be very inefficient and inaccurate. Thus, this paper proposes some possible improvements on these issues, which leads to better simulating results.

THE VRPTW PROBLEM AND MODEL

The basic model of VRP describes a central warehouse with a number of k vehicles of capacity q_k. These vehicles have to transport goods to D locations, represented by 1, 2,…D. The VRP aims to find the shortest distance vehicles have to travel altogether to transport goods to all required locations.

The VRPTW problem can be represented by a weighted graph$G = (V, E)$. $V = \{0,1,2,3 \ldots D\}$ is the node set of distribution center and distribution positions.$E = \{()| \in\neq j\}$ is an arc. If a vehicle goes from node i to node j, then $x_{i+1j+1} = 1$, or else

$x_{i+1j+1}=0$, then the weighted graph can be represented by matrix X: $X=(x_{ij})_{n\times n}, n=D+1$, the element x_{ij} is the path from node $i-1$ to node $j-1$, d_{ij} denotes the distance from node $i-1$ to node $j-1$. If delivery service to node i is done by vehicle k, then $y_i^k=1$, or else $y_i^k=0$. ω_i is the amount of delivery demand at node i; $[e_i,l_i]$ is the time window of node i; s_i is the time for a vehicle to reach node i; a_1 is the vehicle cost; a_2 is the unit distance cost; a_3 is the early arrival punishment coefficient; a_4 is the late arrival punishment coefficient.

Objective Function below:

$$min\, Z=a_1K+a_2\sum_{i=1}^{n}\sum_{j=1}^{n}d_{ij}x_{ij}+\sum_{i=1}^{n}(a_3\, min(s_i-e_i,0)+a_4\, max(s_i-l_i,0))$$

Conditions:

$$\begin{cases} x_{ij}=0, \quad i=j; \\ \sum_{i=1}^{n}x_{i0}=\sum_{j=1}^{n}x_{0j}=K; \\ \sum_{i=1}^{n}x_{im}=\sum_{j=1}^{n}x_{mj}=1, m=2,3,4\dots n; \\ \sum_{i=1}^{D}y_i^k\omega_i\le q_k, k=1,2,3\dots K \\ a_1\gg a_2\gg a_3,a_4 \end{cases}$$

The first function represents that the route from node i to node i is 0; the second function represents that the k-th vehicle from the delivery center that eventually returns to the center; the third function represents that each delivery location is served once; the fourth function represents that the weight of delivery on vehicle k does not exceed its capacity; the fifth function represents that the vehicle costs far outweigh unit distance cost, which far outweigh the punishment coefficients.

USING TRADITIONAL PSO TO SOLVE THE VRP

The traditional Particle Swarm method contain a particle swarm system with M particles $X=\{X_1,X_2,\dots,X_M\}$, $X_i\in R^D$ .The position and velocity of particle i at time k is: $X_i(k)=(X_{i1}(k),X_{i2}(k),\dots,X_{iD}(k))$ and $V_i(k)=(V_{i1}(k),V_{i2}(k),\dots,V_{iD}(k))$, $(i=1,2,\dots,M)$. The hitherto optimum position of X_i is $pbest_i$ .The overall optimum position of X is *gbest*. In the iterative process, the position and velocity update formulas are:

$$V_{ij}^{t+1}=\omega V_{ij}^{t}+f\left(pbest_j^t,gbest_j^t\right) \tag{1}$$

$$X_{ij}^{t+1}=X_{ij}^{t}+V_{ij} \tag{2}$$

$$f\left(pbest_j^t,gbest_j^t\right)=c_1r_1\left(pbest_j^t-\right.$$
$$Xijt+c2r2gbestjt-Xijt \tag{3}$$

$f\left(pbest_j^t,gbest_j^t\right)$ is the function of $pbest_i$ and *gbest*, deciding the direction of particle's motion; ω is the inertia weight; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are random numbers evenly distributed between (0, 1) .

Traditional PSO usually takes the method in document [Zhu, *et. al.*, 2006] to solve VRPs. It constructs a space of dimensions 2N to correspond with VRPs with D delivery missions. Every 2D particle X contains two L-dimension vectors, representing the vehicle X_p that matches each mission and the sequence of each mission X_r in the delivery route of their corresponding vehicle.

For Example: In a VRP with 6 delivery locations and 3 vehicles, the location vector of a particle X can be denoted as:

Delivery Locations: 1, 2, 3, 4, 5, 6

X_p: 2 2 1 1 1 3

X_r: 3.2, 5.4, 1.3, 0.5, 0.7, 2.5

The corresponding delivery route of particle X_i:

Vehicle 1:

Vehicle 2: $0\to1\to2\to0$

Vehicle 3: $0\to6\to0$

Through matching delivery points with vehicles, this representation method removes lots of invalid solutions and decreases the range of the solution to a degree. However, the number of dimensions of particles have increased, so the procedure is not really simplified. When dealing with large-scale VRPs, this algorithm does not effectively lead to the optimum route.

IMPROVING THE PSO AND APPLYING IT ON THE VRP

4.1 Logic of Algorithm

This paper uses the Tabu Search method, using the particle moving direction function $f\left(pbest_j^t,gbest_j^t\right)$ to traverse all the delivery positions from the delivery center. The tabu list is φ. Particle X_i is a sequence of delivery locations $X_i=(X_{i1},X_{i2},\dots,X_{iD})$ representing a delivery route. When the vehicle reaches its maximum capacity, it returns to the delivery center and then continues on its path.

The particle moving direction function of the improved PSO:

$$\begin{aligned} f\left(pbest_j^t,gbest_j^t\right) &= P_1c_1r_1\left(pbest_j^t-X_{ij}^t\right) \\ &+ P_2c_2r_2\left(gbest_j^t-X_{ij}^t\right) \\ &+ P_3\min_j(D)\,, \quad j\in\varphi \end{aligned}$$

P_1, P_2,and P_3 satisfy the functions:

$$(P_1, P_2, P_3) = \begin{cases} (1,0,0)\mu \in [0,0.3) \\ (0,1,0)\mu \in [0.3,0.6) \\ (0,0,1)\mu \in [0.6,1] \end{cases}$$

μ is a random number between [0,1]; P_1, P_2, P_3 are random probability coefficients, driving particles to move towards the best known solution, local best known solution, or the closest position in integer steps； $j \in \varphi$ represents that the next delivery position does not belong to the tabu list； $c_1 = c_2 = 1$； $r_1 = r_2 = 1$; $min_j(D)$ is the closest delivery location X_{ij+1} to the nearest location X_{ij} . This method effectively alters a real-solution space into an integer-solution space, transforming the optimization for the routes between delivery routes into a PSO.

For instance: for a VRP with 6 delivery locations and a maximum vehicle capacity of 10, the particle's location vector X is represented as：

Delivery Locations：1， 2， 3， 4， 5， 6

Demand：4 2 1 5 1 6

Particle position $X_i(k)$: [431265]

Corresponding route： $0 \to 4 \to 3 \to 1 \to 0 \to 2 \to 6 \to 5 \to 0$

$pbest_j$: [234165]

Corresponding route： $0 \to 2 \to 3 \to 4 \to 0 \to 1 \to 6 \to 0 \to 5 \to 0$

$gbest$: [132456]

Corresponding route： $0 \to 1 \to 3 \to 2 \to 0 \to 4 \to 5 \to 0 \to 6 \to 0$

Particle position update $X_i(k)$: $\left[\overset{P_1}{\Rightarrow} 2 \overset{P_1}{\Rightarrow} 3 \overset{P_3}{\Rightarrow} 4 \overset{P_2}{\Rightarrow} 5 \overset{P_2}{\Rightarrow} 6 \overset{P_2}{\Rightarrow} 1\right]$

4.2 Procedures

Step 1： Set the precision requirement, maximum number of iteration, inertia weight and learning factor.

Step 2： Initialize particle position, randomly create N D-Dimension particles $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$， N is the number of particles， D is the number of delivery locations and X_i is a sequence of delivery locations. Every particle X_i represents a unique delivery route, returning to the delivery center when maximum capacity is reached.

Step 3： Set the initial adaptive value as the individual best known solution $pbest_i$ ， and find the population best solution *gbest* by comparison.

Step 4： Repeat the following procedures， until the precision requirement or the maximum number of iteration is met. Iterate the *i*th particle $X_i(k)$ to $X_i(k+1)$. The moving direction of the particle satisfy $f\left(pbest_j^t, gbest_j^t\right)$ —moving from one delivery location to the next based on the random possibility P_1, P_2, P_3.

1. Using the Tabu Search, traverse all delivery locations non-repetitively and update the *i*th particle X_i.
2. Use the adaptive value function to evaluate all of the updated particles $\{X_1(k+1), X_2(k+1), \ldots X_D(k+1)\}$. Update the best known solution $pbest_i$ of every particle and the population best known solution *gbest.*

EXPERIMENT RESULT AND ANALYSIS

This paper uses Matlab 2014 to solve the traditional PSO, Ant Algorithm, and improved PSO for VRP. The performance of the three algorithms are then compared in the same operating system. The results are obtained using the Solomon Data Set to initialize the position and velocity of particles. The Solomon Data Set is as follows:

Table 1 Comparison of different algorithms

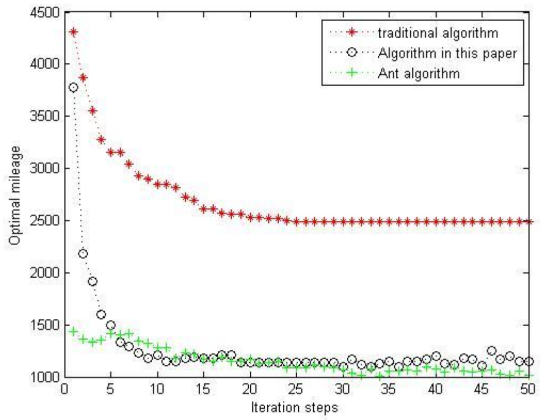
Data Type	Delivery Center	Number of Vehicles	Customer Demand	Node Distribution
C1	（40,50）	200	0-50	clustering
C2	（40,50）	700	0-50	clustering
R1	（35,35）	200	0-41	Random
R2	（35,35）	1000	0-41	Random
RC1	（40,50）	200	0-40	Combined
RC2	（40,50）	1000	0-40	Combined

Initial Parameters： Population Size N=12， Maximum Iteration steps M=50， Unit cost $a_1 = a_3 = a_4 = 0, a_2 = 1$ ， $\omega = 0.5$ ， $c_1 = 0.25, c_2 = 0.75$. The initial position and velocity of the particles are created randomly and the Ant Algorithm is set up with the parameter settings in document [Salman, *et. al.*, 2002]. The three algorithms are used to solve the VRP 10 times each, and the best result are as follows:

Table 2 Comparison of the results of different algorithms

Data	Traditional PSO			Improved PSO			Ant Algorithm		
	Number of Vehicles	Total Cost	Time/s	Number of Vehicles	Total Cost	Time/s	Number of Vehicles	Total Cost	Time/s
C101	14	2442	2.71	10	938	13	10	996	67.88
C201	8	2325	1.53	3	689	14.27	3	826	91.33
R101	13	2832	1.49	8	1042	14.31	8	1034	71.29
R201	10	2955	1.60	2	776	14.38	2	858	79.78
RC101	14	3755	1.37	9	1178	14.04	9	1149	79.59
RC201	8	3523	1.33	2	805	13.55	2	893	63.86

The three algorithms are used separately to solve the C101 data set with 50 iterations. The optimum distance is displayed in the graph:



Graph 1 The Optimization Process of Different Algorithms

Based on Graph 1, the improved PSO guarantees the highest vehicle full-capacity rate and thus decreases the number of vehicles needed. Compared with the Ant Algorithm, the improved PSO is simpler to implement and faster at finding a solution. Compared with the traditional PSO, the improved model results in a far more accurate solution. Based on the optimization process in graph 1, the improved PSO leads to a better route through choosing the closest node $min_j(D)$ and converges to the optimum path more quickly. Yet, the algorithm is still likely to fall into the local maximum point. The species diversity should be increased later on to prevent premature convergence.

CONCLUSION

This paper proposes an improved PSO method based on Tabu Search to solve the VRP. Through simulations run on the classic Solomon Datasets and comparison with the traditional PSO method and Ant Algorithm, I arrive at the conclusion that the improved PSO method is simpler to implement, faster at convergence, and yield far more accurate solutions.

REFERENCE

Chao Ma, Chao Deng, Yao Xiong, Jun Wu. An intelligent optimization algorithm based on hybrid genetic and particle swarm algorithms[J].Computer Research and Development. 2013,50(11):2278-2286.

Dantzig G B, Ramser J H. The truck dispatching problem[J]. Management Science,1959,6(1):80-91.

J. Kennedy and R.C.Eberhart. A discrete binary version of the particle swarm algorithm[J].IEEE International Conference on Systems.1997,5(5):4104-4108.

Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]//1997 IEEE International Conference on Systems,Man,and Cybernetics. Computational Cybernetics and Simulation.IEEE,1997.

M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimisation[J].Congress on Evolutionary Computation.2002,3(3):1957.

Marinakis Y, Marinaki M, Dounias G.A hybrid particle swarm optimization algorithm for the vehicle routing problem[J].Engineering Applications of Artificial Intelligence,2010,23(4):463-472.

Na Tian, Jun Sun, Wenbo Xu, Choi-Hong Lai. An improved quantum-behaved particle swarm optimization with perturbation operator and its application in estimating groundwater contaminant source[J]. Inverse Problems in Science and Engineering.2011,19(2):181-202.

R C Eberhart and Y Shi. Comparing inertia weights and constriction factors in particle swarm optimization[J].Congress on Evolutionary Computation.2002,1(5):84-88.

Salman Aramid I,Al-Madani S.Particle swarm optimization for task assignment problem[J].Microprocessors and Microsystems,2002, 26(8):363-371.

Sen Guo, Guihe Qin, et al. Research on the particle swarm optimization algorithm for multi-objective vehicle route problems.[J]. Xi'an Jiaotong University Academic Journal,2016, 50(9):97-104.

Solomon, Arius Algorithms for the vehicle routing and scheduling problems with time window constraints[J].Operations Research,1987,35(2):254-265.

Wang Z, Guo J, Zheng M. Uncertain multi-objective traveling salesman problem[J].European Journal of Operational Research,2015,241(2):478-489.

Xiaoyong Liu, Hui Fu. Research on VRP based on heuristic ant algorithm. Computer Engineering and Application,2011,47(32):246-248.

Yi Li, Baichuan Lu, Chunxu Liu. Research on chaotic particle swarm optimization algorithm for the vehicle route problem[J]. Chongqing Jiaotong University Academic Journal (Natural Science), 2012,31(4):842-845.

Zhu Q, Qian L, Li Y, et al. An improved particle swarm optimization algorithm for vehicle routing problem with time windows[J].Evolutionary Computation. cec. ieee Congress on,2006:1386-1390.