Routing Optimization for Dispatching Vehicles Based on an Improved Discrete Particle Swarm Optimization Algorithm with Mutation Operation*

Haiyan Liu Xuedan Liu Qiang Wang
College of Computer Science and Information Engineering
Guangxi Normal University
Guilin, China
haiyan1l@163.com

Abstract—**The shortcomings of some existing Particle Swarm Optimization (PSO) algorithms are analyzed and an improved discrete PSO algorithm with wheels topology，which updates the particles' positions by using swap operator, swap sequence and partially matched crossover based on the local PSO， is proposed in the present study. A mutation process in which all the particles are reinitialized when the swarm is stagnated is also introduced. This improved discrete algorithm is then applied to solve the vehicle routing problem (VRP) and a comparison of experimental results from it and other methods is conducted. Results show that the proposed algorithm gets better global convergence and is an effective approach in optimizing the VRP.**

Keywords-vehicle route; discrete PSO algorithm; mutation operation

I. INTRODUCTION

Distribution service plays an important role in modern logistic system. The vehicle routing problem (VRP) [1] involves many constraints, users' requirements, and scattered delivery points with diversity of goods that need to be delivered. Therefore, the transportation routing and time impose significant impact on the controlling of inventory costs, transportation costs and economic efficiency. VRP involves the design of routes with minimum-cost for a fleet of vehicles that serve customers with fixed demands and positions. Each customer is served exactly once in each delivery in a fixed route and the total demand of all customers must not exceed the capacity of the delivery vehicle. VRP is an NP problem. It can be solved by heuristic algorithm or artificial intelligence methods, such as a saving method proposed by Clarck and Wright [2], double population genetic algorithm [3], simulated annealing algorithm [4] and so on.

Since 1995, Particle swarm optimization (PSO) [5] has been widely used to solve continuous optimization problems, but rarely in the discrete issues [6], especially in VRP. In this paper, an improved discrete PSO algorithm is proposed and applied in VRP. Experimental results show that the improved algorithm gets better global convergence and is more effective in optimizing the VRP.

* Project (No 0640067) supported by the Guangxi Nature Science Foundation, Guangxi, China

II. IMPROVED DISCRETE PSO

PSO is an evolutionary computational technique developed by Kennedy and Eberhart in 1995. In classical PSO [9], the particles are manipulated according to the following equations:

$$V_{id}(t+1)=w*V_{id}(t)+c_1*rand()*[P_{id}(t)-X_{id}(t)]+c_2*Rand()*[P_{gd}(t)-X_{id}(t)] \quad (1a)$$

$$X_{id}(t+1)=X_{id}(t)+V_{id}(t) \quad 1 \le i \le s \quad 1 \le d \le N . \quad (1b)$$

In the equations, N is the solution space, s the swarm size and X_i the ith particle's current position. P_i is the previous best position of the ith particle recorded and P_g is the global best position found by any particle. V_i and w denote the current velocity and inertia weight respectively. c_1 and c_2 are two positive constants, and rand () and Rand () are two random functions in the range [0, 1].

The classical PSO is a powerful method to find the minimum of numerical functions, but rarely used in the discrete functions. So, some discrete PSO algorithms illustrated by the traveling salesman problem (TSP) are proposed in [12] and [7] in which:

- Swap Operator (SO): X is the particle's position. Here SO (i_1, i_2) means exchanging $X(i_1)$ with $X(i_2)$. $i_1, i_2 \in$ (1... N). N is the number of the city.
- Swap Sequence (SS) and Velocity: a Swap Sequence consists of one or more SOs. Velocity is a Swap Sequence and defined by $V=\{SO(i_k, j_k)\ldots\}$, $i_k, j_k \in$ (1... N), k=1…$\|V\|$. $\|V\|$ is the number of SO.
- Position plus Velocity: $X'=X+V$ means applying all the SOs in V to X, then obtain X'. X' and X are particle positions while V is velocity.
- Position minus Position: $V=X_2-X_1$ means that the difference between X_2 and X_1 is defined as Velocity V.
- Velocity plus Velocity: $V_3=V_1 \oplus V_2$. V_3 is a SS which contains the SOs of V_1 first, then followed by the SOs of V_2.
- Coefficient multiplied by velocity: c is a real coefficient. So, there are different cases, depending on the value of c:
  Case c=0, then $cV=\Phi$;
  Case $c \in$ [0, 1], then truncate V. The truncated length is the greatest integer smaller than or equal to $c\|V\|$;

978-0-7695-3899-0/09 $29.00 © 2009 IEEE
DOI 10.1109/WGEC.2009.98

Case c>1, it means $c=q+c'$, $q \in N^*$, and $c' \in [0, 1]$, so $cV = \underbrace{V \oplus V \oplus V \oplus \ldots \oplus V}_{q \text{ times } V} \oplus c'V$.

Considering equation (1) is not suitable for discrete PSO algorithm. So, it is rewritten as follows [12]:

$$V_i(t+1)=\alpha V_i(t) \oplus \beta [P_i(t)-X_i(t)] \oplus \gamma[P_g(t)-X_i(t)] \quad (2a)$$

$$X_i(t+1) = X_i(t) + V_i(t) \; 1 \le i \le s, \quad (2b)$$

where α, β and γ are positive constants.

A. Improved discrete PSO

Discrete PSO algorithms in [6] and [8] were introduced to find the solutions to TSP and VRP. However, they are global PSO and easily converge to the local best.

Thus, an improved discrete PSO algorithm is proposed in this paper. We use the local PSO with wheels topology [11], Swap Operator, Swap Sequence and Partially Matched Crossover (PMX) to redefined equation (2a) and (2b) as shown in the following:

$$X_i^1 = X_i(t) + \alpha\, Vi(t) \quad (3a)$$

$$X_i^2 = \beta \text{ PMX } [X_i^1, P_i(t)] \quad (3b)$$

$$X_i(t+1) = \gamma \text{ PMX } [X_i^2, Pl_i(t)], \quad (3c)$$

where α, β and γ are positive constants. They are usually in the range of [0, 1]. Pl_i is the ith particle's neighborhood best position. Equation (3b) means that X_i^1 makes the PMX with $P_i(t)$, where the crossing length is c_1*N (N is the scale of the problem). However, it is different from the genetic algorithm. Here, it is the X_i^1 to make PMX and $P_i(t)$ does nothing. Equation (3c) also does the similar operation and the crossing length is c_2*N. In this way, the position $X_i(t+1)$ of next generation particles is obtained.

In global PSO, all particles exchange information with one particle named P_g. However, in local PSO, particles have information of their own and their neighborhood bests respectively. Therefore, the local PSO increases the diversity of the algorithm to avoid premature convergence.

In wheels topology, the neighborhood of each particle is decided by the subscript and unrelated to the specific location of particles. The neighborhood size in this study is set as 2 regardless of the swarm size. Experimental results show that it gets higher convergence rate. The neighborhood best position is calculated by the following method:

- If the ith particle's subscript is less than or equal to $L/2$ (L is neighborhood length), then choose the best particle from the neighborhood $[1, i+L/2]$. If it is better than Pl_i, then update Pl_i with it.
- If the ith particle's subscript is greater than or equal to $(s-L/2)$ (s is the swarm size), then choose the best particle from the neighborhood $[i-L/2, s]$. If it is better than Pl_i, then update Pl_i with it.
- Else, choose the best particle from the neighborhood $[i-L/2, i+L/2]$. If it is better than Pl_i, then update Pl_i with it.

From (3), it is obvious that if the neighborhood's best position Pl stops moving for long, then the whole swarm stops. It means that all the particles have converged on the best position P_g discovered so far by the swarm. This leads to a premature convergence of the algorithm. To deal with it, the idea of Multi-start PSO proposed in [10] is adopted here. Through initializing all the particles when the algorithm converges to a local minimum, the improved algorithm in this paper can enhance the global search capability and avoid premature convergence.

III. Application of The Improved Discrete PSO in VRP

In VRP, it is desired to design a set of minimum-cost routes for a fleet of vehicles that serve scattered delivery points with fixed positions and certain demands. Each customer is served exactly once and only by one vehicle. Each vehicle has only one route that contains one or more delivery points. All delivery vehicles return to the distribution center at the end. All customers' demands must be met and the sum of their demands is not in excess of the total capacity of all vehicles.

A. Mathematic model for VRP

Assume that a number of vehicles v_s (s=1, 2,...m) with capacity C (here the same capacity is used) are available, Q_i is the loads required to be delivered to points q_i (i=1, 2… k) from distribution center q_0. D_{ij} represents the transportation cost such as distance, time and other costs occurred between q_i and q_j (i, j =0, 1, 2,…, k). Then, some variables are defined as following:

$$x_{sij} = \begin{cases} 1, & \text{if vehicle } s \text{ from } i \text{ to } j \\ 0, & \text{else} \end{cases}$$

$$y_{si} = \begin{cases} 1, & \text{if the demands of } i \text{ is delivered by vehicle } s \\ 0, & \text{else} \end{cases}.$$

Thus, the mathematical model can be expressed as following:

$$\min Z = \sum_{s=1}^{m} \sum_{i=0}^{k} \sum_{j=0}^{k} D_{ij} x_{sij} \quad (4)$$

$$\sum_{i=1}^{k} Q_i y_{si} \le C, \quad s = 1,2,\ldots,m \quad (5)$$

$$\sum_{s=1}^{m} y_{si} = \begin{cases} 1, & i = 1,2,\ldots, k \\ m, & i = 0 \end{cases} \quad (6)$$

$$\sum_{i=0}^{k} x_{sij} = y_{si}, \; j = 0,1,2,\ldots,k; \; s = 1,2,\ldots,m \quad (7)$$

$$\sum_{j=0}^{k} x_{sij} = y_{si}, \; i = 0,1,2,\ldots,k; \; s = 1,2,\ldots,m \quad (8)$$

$$x_{sij} = 0 \text{ or } 1, i, j = 0,1,\ldots,k, s = 1,\ldots,m \quad (9)$$

$$y_{si} = 0 \text{ or } 1, i = 1,\ldots,k, s = 1,\ldots,m \;. \quad (10)$$

Equation (5) is the constraint for vehicle capacity. Equation (6) ensures that the delivery task for each point is served only by one vehicle and all transportation tasks are completed by vehicles m. Equation (7) and (8) limit the arrival and departure of vehicle at a delivery point to one.

B. Constraints

VRP is an integer-programming problem with constraints. Equation (5) describes the constraints for vehicle capacity. The penalty function is introduced here to deal with this constraint, in which a very large positive number R is used as a penalty coefficient, and the objective function is converted to equation (11) by adding the result of multiplying R with the total amount of overload to the objective function:

$$\min Z = \sum_{s=1}^{m}\sum_{i=0}^{k}\sum_{j=0}^{k} D_{ij}x_{sij} + R\bullet\sum_{s=1}^{m}\max\left[\sum_{i=1}^{k}Q_i y_{si} - C, 0\right]. \quad (11)$$

Therefore, the infeasible solutions are given a large fitness value and the swarm converges gradually to the feasible solution in the iteration of PSO.

C. *Integer encoding*

The method introduced in [8] is adopted in this study to describe vehicles' routes. Assume that the number of customers and vehicles are k and m. Then m-1 zero will be inserted into the customer sequence. Therefore, the customer sequence is divided into m parts and each part represents one vehicle's route. In the PSO, every particle corresponds to k+m-1 dimensions.

For example, suppose there are 6 customers and 3 vehicles. The particle's position(X) can be described as: 5 3 0 6 1 4 0 2. It stands for the following routes:

The 1st vehicle: 0 5 3 0

The 2nd vehicle: 0 6 1 4 0

The 3rd vehicle: 0 2 0

This encoding approach converts VRP to a TSP, hence, the traditional methods used in TSP can be used to solve VRP.

IV. EXPERIMENT AND DISCUSSION

In order to illustrate the efficiency of this improved algorithm, experimental results with an example introduced in [3] are also discussed in this paper. The example can be described as following:

There are 8 delivery points and 1 distribution center. The delivery service is offered by 2 vehicles. The total capacity for a vehicle is C=8 tons. Demands Q_i(i=1, 2,…, 8)and distance D_{ij} (i, j=0,1,…,8) are listed in table Ⅰ. Two minimum-cost routes for these two vehicles are to be determined.

TABLE I. DEMANDS AND DISTANCE

D_{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
Q_i (tons)	1	2	1	2	1	4	2	2	

A. *Result of the improved discrete PSO*

The improved PSO proposed in this paper is used in the VRP mentioned above. Every particle in the swarm has 8+2-1=9 dimensions. It represents the arrangement of the vehicle paths, where swarm size is 60, α=0.7, β=γ=0.4, and neighborhood size is 2. *Maxstep* is the unchanged number of consecutive threshold which has a value of 5. Also, Penalty coefficient R=10^8 and the number of iteration is 50. A consecutive experiment with 20 groups of test containing 20 calculations in each was carried out. The optimized results (the shortest distance is 67.5) obtained for each group and their averages are shown in table Ⅱ.

Table Ⅱ shows that the optimal solution obtained with the proposed PSO algorithm reach 19 times with an average of 67.575. The best route is 0 4 7 6 0 and 0 1 3 5 8 2 0.

Fig.1 shows the changes of the fitness values of the best particle in the iteration. As shown in Fig.1, the improved discrete PSO algorithm converges more rapidly to the optimal solution compared with the integer encoding PSO in [8]. Because of the mutation operation, the algorithm is able to jump out of the local best solution and approach the global optimal solution.

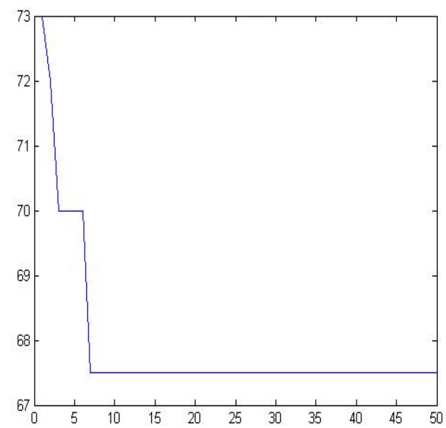

Figure 1. The change of fitness values of the best particle

B. *Comparative Analysis*

A comparison of results from the proposed algorithm and those from references [3] and [8] is conducted. The standard Genetic Algorithm is applied in VRP with swarm size = 60, crossover probability 0.8 and mutation probability 0.05. Double Populations Genetic Algorithm introduced in [3] is also used in VRP with swarm size = 30, crossover probabilities 0.7 and 0.8, and mutation probabilities are 0.05 and 0.1. The parameters of integer encoding PSO algorithm in [8] are: Swarm size is 60, w=0.4, c_1=1 and c2=1.49. Twenty times of calculations were conducted for each algorithm with 50 Iterations. The best results for each method are compared and listed in table Ⅲ.

As shown in table Ⅲ, the standard Genetic Algorithm has an average of 73.25 with no best solution. The Double Populations Genetic Algorithm has an average of 69.575 with 4 times of obtaining the best solution. The average of integer encoding PSO algorithm in [8] is 69.075 with 7 times of getting the best solution. However, the improved algorithm proposed in this study has an average of 67.575 with 19 times of obtaining the best solution. It is obvious that the improved PSO algorithm has a converge rate of 95% in 20 times calculation under the same population size and iterations. It is more efficient than the other algorithms. This improved PSO also converges faster than the integer encoding PSO in [8]. Results show that the improved algorithm is an effective approach in finding the best solutions to VRP.

V. CONCLUSION

For delivery services, reasonable route arrangement is important in improving service quality, enhancing economic efficiency and reducing distribution costs. This study proposes an improved discrete PSO algorithm that combines local PSO, wheels topology, PMX and mutation to find the best solution to VRP. The rate of global convergence of the algorithm proposed in this study is improved remarkably. Experimental results show that the improved discrete PSO algorithm is an effective way in solving discrete combination optimization such as VRP.

REFERENCES

[1] G. Dantzing and J. Ramser, "The truck dispatching problem," Management Science, vol.10, 1959, pp.80-91.

[2] G.Clarke and J.Wright, "Scheduling of vehicles from a central depot to number of delivery points," Operations Research, vol. 12, 1964, pp.12-18.

[3] Yanwei Zhao and Bin Wu, "Double populations genetic algorithm for vehicle routing problem," Computer Integrated Manufacturing Systems, vol. 10, 2004, pp. 303-306.

[4] Guoping Xu and Bin Wu, "Genetic algorithm based on simulated annealing for vehicle routing optimization," Industrial Control Computer, vol. 17, 2004, pp. 49-50.

[5] J. Kennedy and R C.Eberhart, "Particle swarm optimization," Conf, In: Proc IEEE International Conference on Neural Networks, IV Piscataway, NJ: IEEE Service Center, 1995, pp. 1942-1948.

[6] Jianmei Xiao, Junjun Li, and Wang Xihuai, "A modified particle swarm optimization for Traveling Salesman Problems," Computer Engineering and Applications, vol. 40, 2004, pp. 50-52.

[7] Lan Huang and Kangping Wang, "Particle swarm optimization for traveling salesman problems," Journal of Ji Lin University (Science Edition), vol. 41, 2003, pp. 477-580.

[8] Jianmei Xiao and Youfang Huang, "Vehicle routing problem based on discrete particle swarm optimization," Systems Engineering, vol. 23, 2005, pp. 97-100.

[9] Y.SHI and R C.Eberhart, "Particle swarm optimization: developments, applications and resources," Conf, In: Proc Congress on Evolutionary Computation 2001 NJ: Piscataway, IEEE Press, 2001, pp. 81-86.

[10] F.Van Den Bergh, "An analysis of particle swarm optimizer," [D], PhDthesis. Department of Computer Science, University of Pretoria, South Africa, 2002.

[11] Y J. Kenned, "Small worlds and mega-ninds: effects of neighborhood, topology on particle swarm performance," Conf, In: Proceeding of the Congress on Evolutionary Computation, Washington DC, USA, July 1997, pp.1931-1938.

[12] M. Clerc, "Discrete particle optimization illustrated by the traveling salesman problem", 2000

TABLE II. THE TIMES OF BEST SOLUTION AND AVERAGES FOR EACH GROUP OF TEST

Group	1	2	3	4	5	6	7	8	9	10
Times of best solution	16	13	18	13	15	12	17	19	19	14
Averages	67.825	68.075	67.650	68.075	67.875	68.100	67.825	67.575	67.575	68.000
Group	11	12	13	14	15	16	17	18	19	20
Times of best solution	15	16	17	17	17	16	19	14	13	17
Averages	67.875	67.800	67.825	67.725	67.825	67.850	67.575	67.950	68.050	67.750

TABLE III. RESULT COMPARISON

Algorithms \ **Times**	1	2	3	4	5	6	7	8	9	10
Standard genetic algorithm [3]	74 .0	75.0	71.5	72.0	73.5	75.0	76.0	72.5	75.5	73.5
Double populations genetic algorithm [3]	70.0	69.5	67.5	71.0	69.0	70.5	72.0	67.5	71.5	69.0
Encoding PSO algorithm [8]	69.0	70.5	70.0	69.0	70.0	67.5	71.0	67.5	67.5	67.5
The improved discrete PSO	67.5	69.0	67.5	67.5	67.5	67.5	67.5	67.5	67.5	67.5
Algorithms \ **Times**	11	12	13	14	15	16	17	18	19	20
Standard genetic algorithm [3]	72.0	69.0	73.0	75.5	72.0	73.0	75.0	73.5	71.5	75.0
Double populations genetic algorithm [3]	67.5	69.0	71.0	70.0	67.5	70.5	69.0	69.5	71.0	69.0
Encoding PSO algorithm [8]	70.5	67.5	70.0	70.0	69.0	71.0	69.0	70.0	67.5	67.5
The improved discrete PSO	67.5	67.5	67.5	67.5	67.5	67.5	67.5	67.5	67.5	67.5