OPTIMAZATION OF CITY LOGISTICS' DISTRIBUTION ROUTE BASED ON ANT COLONY SYSTEM

GONG Jinli[1] WANG Quan[1]

[1]School of Transportation Engineering, Tongji University, Shanghai, P. R. China, P.O. Box 200092; PH +8621 79584542; e-mail: gjl2001cd@126.com

Abstract: In this paper, the City Logistics' distribution route optimization problem is ascribed as Vehicle Routing Problem with Time Windows (VRPTW) and its mathematical model has been established. Ant Colony System algorithm (ACS) that is capable of searching multiple search areas simultaneously in the solution space is good in diversification. It has been successfully applied to many NP-hard problems. An improved Ant Colony System algorithm (IACS) is proposed in this paper. The initial solution was produced by the nearest neighbor heuristic, then algorithm's efficiency was improved by 2-opt and insertion move. At last, it is realized in Matlab, and its validity has been proved by C101 problem of Solomon benchmark problems. The results also indicate that such an algorithm outperforms the original heuristic.

Keywords: City Logistics, Distribution Route, VRPTW, IACS

1. Introduction

The City Logistics' distribution route optimization problem we discussed can be ascribed as Vehicle Routing Problem with Time Windows (VRPTW), which belongs to NP-hard problem. In the paper, the problem's model has been established with determinate vehicle number. Ant Colony System algorithm (ACS) has been already successfully applied to many problems. The paper has taken it as the way to solve the problem. Based on the general algorithm, an improved Ant Colony System algorithm (IACS) is proposed in this paper. The choosing, updating and cooperating mechanism have been improved by using 2-opt and insertion move in pheromone update. The algorithm has also been realized in Matlab and its validity has been proved by C101 problem of Solomon benchmark problems.

2. Mathematical model

Useful variable and parameters in the model will be defined as follows: b_j^k : the time vehicle k starts service at node j; c_{ij}: travel distance of edge (i,j); d_i: demand of node i, $i \in N$; e_i : the earliest starting time of node i; l_i: the latest starting time of node i; N: the set of nodes, 0 denotes the distribution center; C: capacity of vehicle; t_{ij}: travel time of edge (i,j), and it includes the service time at node i; V: set of vehicles; x_{ij}^k : decision variable, if vehicle k visits node j immediately after node i and $i \neq j$, its value will be 1, else it equals to 0。

Object function:

$$Min\sum_{k\in V}\sum_{i\in N}\sum_{j\in N}c_{ij}x_{ij}^k \tag{1}$$

s.t:

$$\sum_{k\in V}\sum_{i\in N} x_{ij}^k = 1 \qquad \forall j \in N\setminus\{0\} \tag{2}$$

$$\sum_{k\in V}\sum_{j\in N} x_{ij}^k = 1 \qquad \forall i \in N\setminus\{0\} \tag{3}$$

$$\sum_{i\in N} d_i \sum_{j\in N} x_{ij}^k \le Q \qquad \forall k \in N \tag{4}$$

$$\sum_{i\in N} x_{ih}^k - \sum_{j\in N} x_{hj}^k = 0 \qquad \forall h \in N\setminus\{0\}, \forall k \in V \tag{5}$$

$$\sum_{j\in N\setminus\{0\}} x_{0j}^k = 1 \qquad \forall k \in V \tag{6}$$

$$\sum_{i\in N\setminus\{0\}} x_{i0}^k = 1 \qquad \forall k \in V \tag{7}$$

$$x_{ij}^k (b_i^k + t_{ij}) \le b_j^k \qquad \forall i \in N, \forall j \in N, \forall k \in V \tag{8}$$

$$e_i \le b_i^k \le l_i \qquad \forall i \in N, \forall k \in V \tag{9}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall i \in N, \forall j \in N, \forall k \in V \tag{10}$$

The object function (1) is to minimize the total traveled distance or total cost; the constraints (2) and (3) ensure every node can only be severed once by a vehicle which start off from distribution center; constraint (4) means that any vehicle's load can't surpass its capacity; constraint (5) tells us that if vehicle k visits a node then it will also leave from the same node; constraints (6) and (7) suggest that every vehicle should leave from and come back to distribution center; constraint (8) indicates vehicle k from node i can't arrive to node j before the time $b_i^k + t_{ij}$; constraint (9) shows the starting time of service must within the time windows; constraint (10) makes sure variable is a number within the scope [0,1].

3. Improved ants colony system algorithm

3.1 The initialize of the pheromone

In this paper, the initial pheromone is decided by formula (11).

$$\tau_0 = (n * L_{nn})^{-1} \tag{11}$$

Here n is the nodes' number, L_{nn} is the length calculated by the nearest neighbor heuristic (Chen et al. 2004).

3.2 Solution's construction

In the paper, the roulette strategy has been applied to define probability $P_{ij}^k(t)$, by which vehicle k transfer from node i to node j.

$$P_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum\limits_{h \in U} [\tau_{ih}(t)]^\alpha [\eta_{ih}(t)]^\beta} & j \in U \\ 0 & j \notin U \end{cases} \tag{12}$$

Here, U is the set of unvisited nodes. τ_{ij} is the pheromone on line (i,j); α , β separately represents controlling pheromone and heuristic pheromone's rate in $P_{ij}^k(t)$. η_{ij} represents the difference between serving node i and j in one route and in two different tours. According to the shortest route goal, the heuristic parameter η_{ij} will be calculated by the following equation:

$$\eta_{ij} = d_{i0} + d_{0j} - d_{ij} \tag{13}$$

Here, d_{ij} represents the distance from node i to node j; 0 represents the distribution center.

The route's construction principle of IACS designed here is: firstly, generate the starting nodes at random for every ant; secondly, use roulette strategy to choose next node and continuously repeat it until every node has been visited; at last, partition the route according to vehicle's capacity constraint and nodes' time windows.

3.3 Local search

When all ants complete their routes, they will be optimized by local search strategy. However, it often takes much time to do local search in the original ACS. So in the IACS, 2-opt and insertion move are applied in local search to find the best solution. Figure1 shows the process of 2-opt. Fig.2 illustrates insertion move, the selected nodes are inserted to same route (Fig.2a) or different route (Fig.2b) in order to reduce route's length.

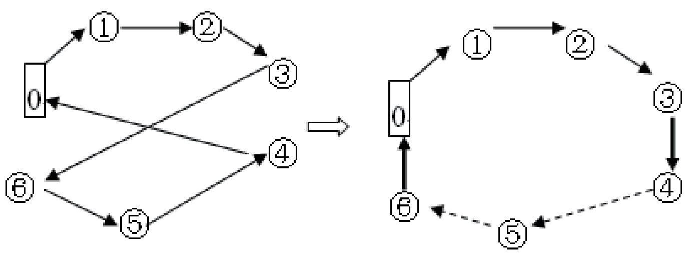
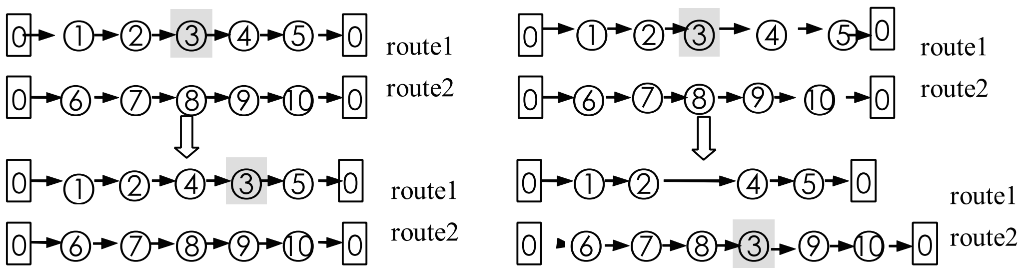


Fig.1 2－opt



(a)Insert in same route (b)Insert in different route

Fig.2 Insertion move

3.4 Update pheromone

The pheromone's update rule is showed as follows:

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho\sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{14}$$

Here, $$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q}{L(k)} & x_{ij}^{k}=1 \\ 0 & x_{ij}^{k}\neq 1 \end{cases} \tag{15}$$

L_k represents the length ant have passed through; Q is a constant, which denotes the sum of pheromones emitted after one complete route search has been finished; x_{ij}^{k} is the same variable as part two. $0 \leq \rho \leq 1$ is a user parameter called volatilization value, reflecting the volatilizing degree of corresponding pheromone; $\Delta\tau_{ij}(t)$ is the pheromone value left by ant when it gets across line (i, j).

3.5 The whole IACS rule

Fig.3 shows IACS's whole flow chart. The step can be described as follows:

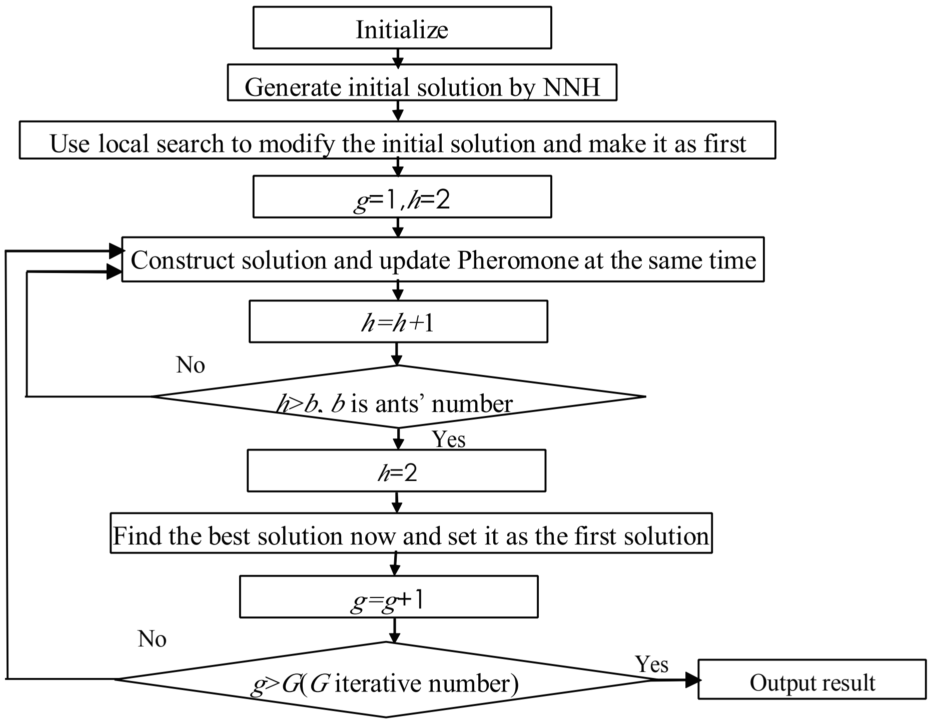


Fig.3 IACS program flow chart

(**1**) Initialize the parameters; (**2**) Generate an initial solution by NNH; (**3**) Use local search (2-opt and insertion move) to modify the initial solution and make it as first solution, set g=1, h=2; (**4**) Construct the route based on construction principle and update the pheromone at the same time. h=h+1; (**5**) If h>b, set h=2 and go to step 6, else, back to step 4; (**6**) Record the best solution so far, and set it as the first solution for next generation, g=g+1; (**7**) If fulfill the stop condition (the stop condition in the paper is achieving the maximum iterative times), stop operation, output the result, else, back to step 4.

4. Instance analysis

In order to prove IACS's validity and compare its calculation ability with other algorithm, we use Matlab to realize its programming, and analyze C101 problem of Solomon (1987) benchmark problems.

4.1 Programming

4.1.1 Parameters setting

We use Gui of Matlab to set parameters and open the data files. The parameter needed set here include: ants number $b=n/10$(n is the number of nodes), $\alpha=1$, $\beta=1$, pheromone volatilization parameter $\rho=0.1$, iterative number $G=n/2$ (Fig.4).

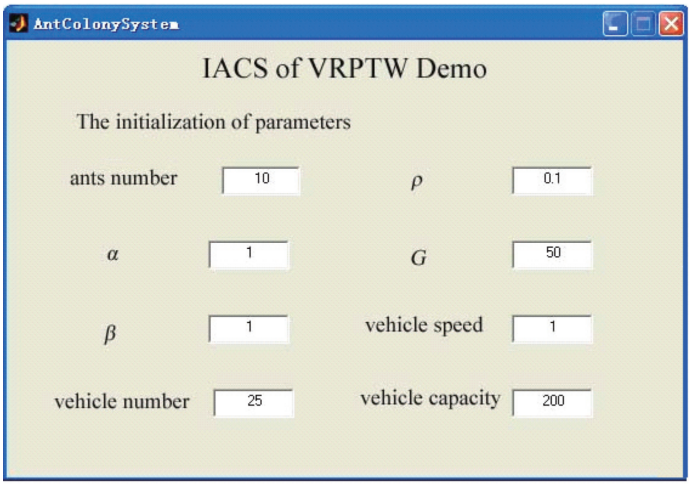


Fig.4 parameters' initialization interface chart

4.1.2Local search

Computational method of object's value has been altered in the paper to satisfy the vehicle capacity limit and the time windows after local search operation. Set route's increasing length as:distance(i,j)+M*(max(dsum-cap,0)+max(time-dt(i), 0)). Here, M is a extremely big number (equal to 10000000 in the paper), dsum represents vehicle's adding load after visiting node j, dt(j) is the allowed latest time. The object value's formulation is: solution=solution+distance(i, j)+M*(max(dsum-cap,0)+max(time-dt(i), 0)).

4.1.3 Solution's construction

The ant find next node by diverting probability, which is showed in equation (12). Set the diverting probability matrix as P. The element P(i,j) represents the probability from city i to city j; matrix Tabu represents the set of nodes ants haven't visited, Tabu(k, j) == 0 denotes ant k hasn't visited node j. Then calculate the value of $(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}$ from node i to all the unvisited nodes and put the value into matrix psum_medium. Use function to calculate their summary, then find maximal one of row i and corresponding value j. Calculate the probability P(i,j), if Tabu(k, j) == 0, $p(i,j)=(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}/psum(k,i)$, else if Tabu(k, j) =1, P(i, j) =0. Find the maximal element from row i in matrix P, and set the corresponding j as the next node ant will visit.

4.2 Instance validation

The c101 of Solomon benchmark problems has been used as the instance to prove algorithm's operation validation, Fig.5 show the final result. The ending computing results of the paper are: the object value is 828.94, vehicle number is 10. Compared with the existed optimum solution: the object value is 827.3, vehicle number is 10 (Desrochers et al. 1992), the paper has got the approximate results.

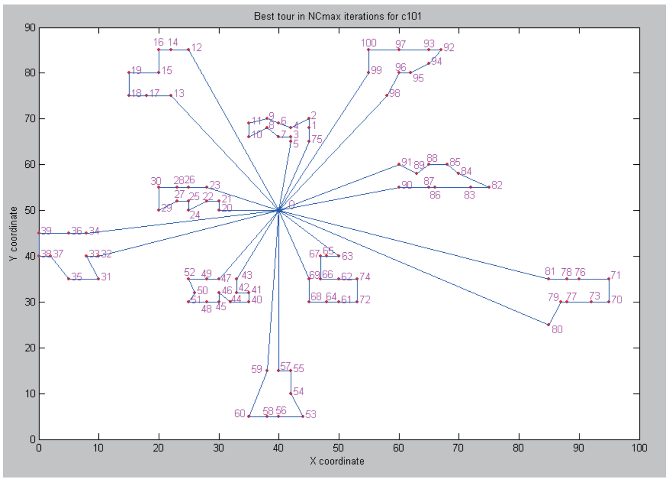


Fig.5 c101's optimum route

5. Conclusion

An improved Ant Colony System algorithm (IACS) is proposed in this paper, and it has been used to optimize City Logisitic's distribution route. It has been realized in Matlab and an instance has also been used to prove its operation validation. However, the object of VRPTW model has not contained vehicle's number. Therefore, how to enhance algorithm should be further studied to make route's total length shorter when the vehicles' scale gets smaller. Moreover, this algorithm should be further applied in the actual problem to validate its usability and compatibility.

Reference:

[1]Chen C.H and Ting C.J. An improved ant colony system algorithm for the vehicle routing problem[J], Working Paper 2004-001, Department of Industrial Engineering and Management, Yuan Ze University, Taiwan, R.O.C. 2004

[2]Wu Jian-Jun, Liu Jun. The hybrid ant colony system of Logistics' distribution route's arranging problem [J]. Construction engineering, 2004, 8：98~101

[3]Dorigo M., Maniezzo V. and Colorni A. Ant system: Optimization by a colony of cooperating agents [J], IEEE Transactions on Systems, Man and Cybernetics Part B, 1996, Vol. 26, 29-41.

[4] Desrochers M., J. Desrosiers, and M. Solomon. A new optimization algorithm for the vehicle routing problem with time windows[J]. Operations Research, 1992, 40(2): 342～354.