Ant Colony System based on the ASRank and MMAS for the VRPSPD

Tao Zhang, Chuoya Yu
School of Information Management and Engineering
Shanghai University of Finance and Economics
Shanghai 200433, China
taozhang@mail.shufe.edu.cn

Wenxin Tian
Information Office in Center of Modern Education Technology
Shanghai Finance University
Shanghai 201209, China

Yuejie Zhang
Department of Computer Science and Engineering,
Shanghai Key Laboratory of Intelligent Information Processing
Fudan University
Shanghai 200433, China
yjzhang@fudan.edu.cn

Abstract—**This paper builds a mixed integer programming model for the VRPSPD. An Ant Colony System (ACS) approach combining with the pheromone updating strategy of ASRank and MMAS ant algorithm is proposed. The initial vehicle load is designed to be a random value correlated to the delivery and pick-up demands of the rest clients. The experimental results show that the approach could improve the vehicle load rate and get rid of the added total distance caused by fluctuating load and maximum capacity constraint. It could obtain the satisfied solution with high convergence speed in an acceptable time.**

Keywords-VRPSPD; ant colony system; mixed integer programming

I. Introduction

VRPSPD (Vehicle Routing Problem with Simultaneous Pickup and Delivery) was first introduced by Min[1], where book distribution and recollection activity between a central library and 22 local libraries with limited number of vehicles and available capacity is discussed. Gendreau et al.[2] studied VRPSPD with only one vehicle by first solving a TSP then arranging the order of delivery and pick-up in the TSP route. Tang & Galvao proposed two local search heuristics to solve TSPSPD, the first of which is an adaptation of a tour partitioning heuristic and the second uses the sweep algorithm[3]. Dethloff[4] first studied VRPSPD from the point of view of reverse logistics. Tang & Galvao first proposed a mathematical model of VRPSPD with the maximum travel distance constraint, and solved the problem using taboo search and a mixed local search heuristics[5].

Anily[6] proved that VRPB is NP-hard, since VRPB may be considered as a special case of VRPSPD when either the delivery demand or pick-up demand of each client equals zero[7], VRSPD is also a NP-hard. The accurate algorithms can't obtain satisfied solutions of the large size VRPSDP in valid time. The ant colony algorithm is one of the parallel algorithms proposed for a variety of combinatorial optimization problems, with good searching performance of discovering better solution and accelerating the evolution process. Biilent and Elif[8] first improved ACS algorithm to solve the capacitated VRPSPD, introducing savings heuristics into visibility function and using 2-opt local heuristics.

In this paper we solve VRPSPD with maximum vehicle travel distance (ML_VRPSPD) using an improved ACS algorithm. In section 2, a mixed integer programming model for ML_VRPSPD is established. Section 3 is devoted to the improved ACS algorithm for ML_VRPSPD. Section 4 discusses the experimental analysis.

II. Problem Description and Modeling

The problem tackeled in this paper deals with a single depot distribution/collection system servicing a set of clients by means of a homogeneous fleet of vehicles with the same capacity and maximum travel distance. Each client has both of a delivery demand and a pick-up demand and the size of the picked up and delivered goods are identical, consuming the same amount of capacity on each vehicle. The vehicles deliver goods to clients from the depot and simultaneously pick-up loads back to the depot at the end of their trip while satisfying the capacity and the maximum distance constraints. The objective is to find the set of routes servicing all the clients at the minimum cost (total travel distance).

Notation:

n denotes the total number of clients;

C denotes the set of clients, $C = \{1,2,3,...,n\}$;

V denotes the set of vehicles;

A denotes the set of clients plus depot, $A = \{0\} \cup C$, where 0 represents the depot;

Q denotes the vehicle capacity;

This paper is supported by National Natural Science Fund of China (No.70501018, 70602031) and "211 Project" of Shanghai University of Finance & Economics.

L denotes the maximum travel distance of vehicle;

c_{ij} denotes the distance between client i and j ($i \in A,\ j \in A,\ i \neq j$; $c_{ii} = \infty (i \neq 0)$; $c_{00} = 0$);

d_i denotes the delivery demand of client i;

p_i denotes the pick-up demand of client i;

q_{ijk} denotes the vehicle load after visiting client i and before visiting client j.

Decision variable:

$$x_{ijk} = \begin{cases} 1, & \text{if arc } (i,j) \text{ belongs to the route operated by vehicle } k \\ 0, & \text{otherwise} \end{cases}$$

The corresponding mathematical formulation is given by

$$min \sum_{k \in V} \sum_{i \in A} \sum_{j \in A} c_{ij} \cdot x_{ijk} \tag{1}$$

$$\sum_{i \in A} \sum_{j \in A} c_{ij} x_{ijk} \le L \ , \qquad \forall k \in V \tag{2}$$

$$\sum_{k \in V} \sum_{i \in A} x_{ijk} = 1 \ , \qquad \forall j \in C \tag{3}$$

$$\sum_{i \in A} x_{ijk} - \sum_{i \in A} x_{jik} = 0 \ , \qquad \forall j \in A, \forall k \in V \tag{4}$$

$$\sum_{i \in A} x_{0ik} \le 1 \ , \qquad \forall k \in V \tag{5}$$

$$\sum_{j \in A} q_{0jk} = \sum_{i \in A} \sum_{j \in A} x_{ijk} \cdot d_j \ , \qquad \forall k \in V \tag{6}$$

$$\sum_{k \in V} \sum_{i \in A} q_{ijk} - d_j = \sum_{k \in V} \sum_{i \in A} q_{jik} - p_j \ , \ \forall j \in A \tag{7}$$

$$\sum_{i \in A} q_{i0k} = \sum_{i \in A} \sum_{j \in A} x_{ijk} \cdot p_j \ , \qquad \forall k \in V \tag{8}$$

$$q_{ijk} \le Q \ , \qquad \forall i, j \in A, \forall k \in V \tag{9}$$

$$q_{ijk} \ge 0 \ , \qquad \forall i, j \in A, \forall k \in V \tag{10}$$

The objective function (1) seeks to minimize total distance traveled. Restriction (2) is the maximum travel distance constraint; constraint (3) ensures that each client is visited by exactly one vehicle; constraint (4) guarantees that the same vehicle arrives at and departs from each client it serves; constraint (5) guarantees that each vehicle can exactly be used once; constraint (6) guarantees that each vehicle leaves the depot fully loaded with the goods to be distributed while the pick-up load is null; constraint (7) is flow equation which guarantee that both demands are satisfied for each client; constraint (8) guarantees that when vehicles return back to the depot ,they have distributed all their deliveries and are fully loaded with the picked up quantities; restriction (9) imposes an upper limit on the total load transported by a vehicle in any given section of the route; (10) is non-negativity constraint.

III. The Ant Colony Algorithm for ML_VRPSPD

A. The ant colony optimization

The ant colony algorithm is one kind of simulation evolution algorithms first proposed by Italian scholar M. Dorigo, inspired from the behavior of the real ants in nature looking for food. The biologist discovered from massive research that, the real ant releases one kind of special secretion ----Pheromone on the way it has passed, which enables other ants in the certain scope to sense the existence and the intensity of the pheromone to instruct their own behavior. The ant colony favors to move along the way on which the pheromone intensity is strong, namely greater level of pheromone on a path will increase the probability that ants will follow that path. It is this kind of information exchange among ants that causes the collective behavior of the ant colony to find the best way to get the food.

The ant system (AS) is the earliest ant algorithm[9] and is first used to solve the traveling salesman problem[10]. Many new improved ant algorithms have been gradually developed on the basic ant colony algorithm, for example, Ant Colony System (ACS)[11], MAX-MIN Ant System (MMAS)[12], Rank-based Version of Ant System (ASRank)[13], Multi Colony Ant Algorithm[14] and so on.

B. Construction of heuristic function

In view of the complexity of the ML_VRPSPD problem, this paper improves the construction of heuristic function and the design of the initial delivery load of the vehicles, combining the pheromone update strategy of the ASRank and the MMAS.

The ant colony can make dynamic adjustment from the first phase to the second phase according to the current load as in (11):

$$\eta_1(i,j) = 1/c_{ij} \qquad \eta_2(i,j) = 1/\xi_{ij}$$

$$\xi_{ij} = \begin{cases} load - (d_i - p_i), & load \ge \lambda_1 * start\_load \\ load + (d_i - p_i), & load < \lambda_1 * start\_load \end{cases} \tag{11}$$

Where $load$ is the current delivery load, $start\_load$ is the initial delivery load, and λ_1 is an adjustable parameter, $0 < \lambda_1 < 1$. So the construction of the total information is as follows:

$$total(i,j) = [\tau(i,j)]^{\alpha} \cdot [\eta_1(i,j)]^{\beta} \cdot [\eta_2(i,j)]^{\gamma} \tag{12}$$

Where, $\tau(i,j)$ is the pheromone level between i and j, $\eta_1(i,j)$ is the distance heuristic function, $\eta_2(i,j)$ is the residual loading capacity heuristic function between i and j. α, β, and γ are their corresponding weighted values in the total information.

C. Pseudo-random probability selection rule

Using the pseudo-random probability selective rule, ants will choose the next client according to formula (13).

$$j = \begin{cases} \arg\max_{u \in M_i} \{[\tau(i,u)]^{\alpha}[\eta_1(i,u)]^{\beta}[\eta_2(i,u)]^{\gamma}\} & if\ q \le q_0 \\ s & otherwise \end{cases} \quad (13)$$

Where q_0 is a constant, $0 \le q_0 \le 1$, q is a random number uniformly distributed in $[0,1]$. M_i is the set of feasible clients. Each ant may either visit the most favorable client or randomly select a client based on a probability distribution according to formula (14). $P(i,s)$ is the possibility of the ant taking s as the next client.

$$P(i,s) = \begin{cases} \dfrac{[\tau(i,s)]^{\alpha}[\eta_1(i,s)]^{\beta}[\eta_2(i,s)]^{\gamma}}{\sum_{u \in M_i}[\tau(i,u)]^{\alpha}[\eta_1(i,u)]^{\beta}[\eta_2(i,u)]^{\gamma}} & if\ s \in M_i \\ 0 & otherwise \end{cases} \quad (14)$$

D. Pheromone update rules

Pheromone trail is updated both locally and globally. Once an arc ij is chosen by an ant, its level of pheromone is change by applying the local trail updating formula (15):

$$\tau(i,j) = (1-\rho)\cdot\tau(i,j) + \rho\cdot\tau_0 \quad (15)$$

Where ρ is an adjustable parameter which denotes the pheromone volatility coefficient, $\rho \in (0,1)$. $\tau_0 = 1/(n \cdot T_{nn})$, and T_{nn} is the total travel distance in initial feasible solution generated by the near neighbor heuristics. The ACS algorithm only allows to update the pheromone on the route of the global best solution, while in the global updating rule of ASRank all routes found in one iteration are sorted ascendingly according to the route length.

Once one iteration is finished, the pheromone level is updated by applying the global updating rule of formula (16):

$$\tau(i,j) = (1-\rho)\tau(i,j) + \sum_{r=1}^{w-1}(w-r)\Delta\tau_{ij}^{r} + w\Delta\tau_{ij}^{gb} \quad \rho \in (0,1) \quad (16)$$

Where $\Delta\tau_{ij}^{r} = 1/length^{r}$, $\Delta\tau_{ij}^{gb} = 1/length^{gb}$, $length^{r}$ is the length of the r^{th} best solution, $length^{gb}$ is the length of the global best solution. One pheromone updating in the MMAS algorithm is also adopted to avoid the stagnancy phenomenon in the process of searching: the quantity of pheromone on each arc is limited to the scope $[\tau_{\min}, \tau_{\max}]$, where $\tau_{max} = n\tau_0/\rho$, $\tau_{min} = \tau_{max}/(2n)$. In this way the pheromone level on the tour of each ant will not differ greatly so as to avoid stopping in local best solution too early.

IV. Experimental Analysis

The improved ACS algorithm for ML_VRPSPD is tested on one of the benchmark problem instances of Dethloff[4]. Random 10 groups of test instances SCA3 are generated: the coordinates of 50 clients are uniformly distributed over the interval [0,100]. The delivery demand d_i of clients are uniformly distributed over [0,100], the pick-up demand is computed by using a random number r that is uniformly distributed over $[0,1]$ such that $p_i = d_i * (0.5 + r)$. The maximum capacity of vehicles is generated by the minimal number of vehicle 3, $Q = \sum d_i / 3$. The algorithm is coded using C++, running on Pentium III.

Table 1 displays average results, average iteration times and average computing time of 10 separate experiments on the 10 groups of SCA3 instances and compares the average results of 10 experiments of ML_VRPSPD with those of VRPSPD solved by Biilent and Elif [8] and Dethloff [4].

TABLE I. Comparison of results of 10 sets of results

No.	ML_VRPSPD Best.	ML_VRPSPD Avg.	Time	VRPSPD Best.	VRPSPD Avg.	Dethloff Avg.
1	705	726	30.91	653	666	689
2	740	749	15.94	721	738	765
3	785	791	20.89	685	692	742
4	676	691	28.80	701	708	737
5	683	701	32.63	709	719	747
6	723	729	36.97	716	721	784
7	679	698	33.61	660	670	720
8	738	759	32.94	660	680	707
9	650	653	29.03	754	759	807
10	744	756	18.05	683	693	764
avg	712	725	27.97	695	705	746

From table 1, the mean Avg. of ML_VRPSPD is 725, obviously less than the mean Avg. 746 of Dethloff. The mean Best. 712 and the mean Avg. 725 are more than these 695 and 705 of Biilent and Elif[8], but the mean Best. 712 approaches that of Biilent and Elif. Since the tighter constraints of ML_VRPSPD compared with those of VRPSPD, and 2-opt heuristics using in paper [8], the above differences are acceptable and the improved ACS algorithm is effective and able to find the satisfactory solution in given computing time.

Fig.1 is astringency curve of the objective function obtained in one of the experiments on SCA3-0, the given iteration times is 2,500, the x-coordinate stands for iteration times, the y-coordinate stands for the corresponding total travel distance.

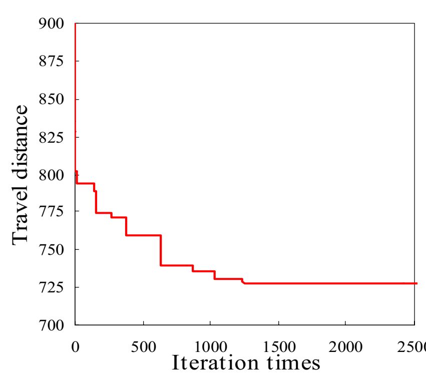

Figure 1. Astringency curve

From Fig.1, the convergence rate within 100 iterations is extremely quick; it slowly decreases with iterative times between 200 and 1,000; finally converges to the best solution of 728 at about 1,500 iterations.

Fig.2 is the vehicles state curve corresponding to the best solution.

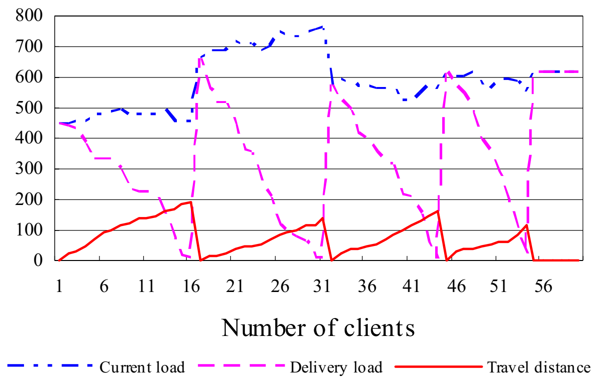

Figure 2. Vehicles states curve

In Fig.2, the delivery load of each vehicle decreases gradually, and the current load of each vehicle fluctuates continuously, while in the latter phase it presents the tendency of undulatory increase. So an effective utilization of vehicle capacity has been made under the maximum capacity constraint. When the vehicles return back to the depot, their residual delivery loads approach to 0. So the vehicles return to the depot because of the lack of delivery goods rather than the load limit. The current travel distance increases with the increasing number of visited clients, and doesn't violate the maximum travel distance of 200.

V. CONCLUSION

In this paper the VRPSPD is solved with maximum travel distance constraint, using an improved ACS algorithm according to the complex feature of fluctuating vehicle load. We also established a mixed integer programming model for ML_VRPSPD. The vehicles residual loading capacity is introduced into the heuristic function, thus the total information contains travel distance together with the current vehicle load. The initial delivery load is designed to be a stochastic value correlated with the demands of unvisited clients. The pheromone update strategy of ASRank and MMAS is used to replace that of ACS. The experimental results indicated that the improved ACS algorithm can enhance the utilization of capacity and reduce the total travel distance avoiding vehicles repeatedly returning to the depot because of dissatisfying the maximum capacity and the maximum travel distance constraints. Experimental results show that the improved ACS algorithm designed in this paper is effective and is able to find the satisfactory solution in acceptable computing time with satisfying convergence rate.

ACKNOWLEDGMENT

Yuejie Zhang is the corresponding author.

REFERENCES

[1] H. Min, "The multiple vehicle routing problems with simultaneous delivery and pick-up points," Transportation Research, 1989, 23A: 377–386.

[2] M. Gendreau, G. Laporte, and D. Vigo, "Heuristics for the travelling salesman problem with pickup and delivery," Computers and Operations Research ,1999,26: 699–714.

[3] F. A. Tang, and R. D. Galvão, "Vehicle routing problems with simultaneous pick-up and delivery service," Journal of the Operatioanl Research Society of India (OPSEARCH), 2002, 39:19-33.

[4] J. Dethloff, "Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up," OR Spektrum, 2001, 23: 79-96.

[5] G. Tang, "A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," Computer & Operations Research , 2006, 33: 595-619.

[6] S. Anily, "The vehicle-routing problem with delivery and back-haul options," Naval Research Logistics, 1996, 43: 415–434.

[7] J. Dethloff, "Relation between Vehicle Routing Problems: an Insertion Heuristic for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Applied to the Vehicle Routing Problem with Backhauls," Journal of the Operational Research Society, 2002, 52: 115–118.

[8] B. Çatay, and Eİ. Gökçe, "An ant colony system approach for the capacitated vehicle routing problem with simultaneous delivery and pick-up," Working Paper, Sabanci University, 2005.

[9] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics–Part B, 1996, 26: 29-41.

[10] M. Dorigo, and L. M. Gambardella, "Ant colonies for the traveling salesman problem," BioSystems, 1997, 43:73-81.

[11] M. Dorigo, and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," IEEE Transactions on Evolutionary Computation 1, 1997, 1: 53-66.

[12] T. Stutzle, and H. H. Hoos, "MAX-MIN ant system," Future Generation Computer Systems, 2000, 16(8): 889-914

[13] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the ant system: A computational study," Central European Journal for Operation Research and Economics, 1999, 7(1):25-28.

[14] M. Middendorf, R. Frank, and S. Hartmut, "Multi colony ant algorithms,"Journal of Heuristics, 2002, 8(3): 305-320.