MATEC Web of Conferences 22, 03022 (2015)
DOI: 10.1051/matecconf/20152203022

Solving the Capacitated Vehicle Routing Problem Based on Improved Ant-clustering Algorithm

Jiashan Zhang*, Xiaoqun Lin & Yi Jun
Chongqing Vocational Institute of Engineering, Jiangjin, Chongqing, China

Qiang Li
College of Business Administration, Chongqing Science and Technology University, Chongqing, China

ABSTRACT: The capacitated vehicle routing problems (CVRP) are NP-hard. Most approaches can solve small-scale case studies to optimality. Furthermore, they are time-consuming. To overcome the limitation, this paper presents a novel three-phase heuristic approach for the capacitated vehicle routing problem. The first phase aims to identify sets of cost-effective feasible clusters through an improved ant-clustering algorithm, in which the adaptive strategy is adopted. The second phase assigns clusters to vehicles and sequences them on each tour. The third phase orders nodes within clusters for every tour and genetic algorithm is used to order nodes within clusters. The simulation indicates the algorithm attains high-quality results in a short time.

Keywords: CVRP; three-phase heuristic; ant-clustering algorithm; genetic algorithm

1 INTRODUCTION

Vehicle routing problems (VRPs) have been the subject of intensive research for more than 50 years, due to their great scientific interest as difficult combinatorial optimization problems and their importance in many application fields, including transportation, logistics, communications, manufacturing, military and relief systems, and so on. Hundreds of models and algorithms have been developed to obtain either optimal or heuristic solutions for different versions of VRP, in which the capacitated Vehicle Routing Problem (CVRP) is one of the most famous and widely studied problems. The CVRP was introduced in the seminal article by Dantzig and Ramser (1959) under the name "Truck Dispatching Problem" [1]. The current name (CVRP)[2] of the problem became widespread in the article by Christofides (1976). The majority of current researches on CVRP focus on the problems within a limited size of 100 customers [3]. However, it is common for real-life vehicle routing applications, such as waste collection, courier service, beverage distribution and milk collection and delivery, to involve the daily service of hundred or even thousand customers.

Lenstra and Rinnoy Kan (1981) have analyzed the complexity of the vehicle routing problem and have concluded that practically all vehicle routing problems are NP-hard because they are not solved in polynomial time [4]. The exact algorithms and traditional heuristic algorithms are difficult, even impossible, to solve CVRP. First, the distance in a straight line isn't able to meet problem any longer. Second, calculating the distance matrix is time-consuming. Actually, besides the distance between customers and the distribution center, the distances among adjacent customers are required, whereas customers away from each other usually don't belong to the same distribution route and there is little probability of using them. That's to say some (not all) of the distance matrixes are used in the process of calculating. So calculating all the distances between customers are unnecessary.

In this paper, the solution is attained through three-phase heuristic, which first involves the conversion of CVRP to TSP, using ant-clustering algorithm and then generate tour and improve it.

The rest of this paper is organized as follows. Section 2 introduces the relevant literature. A mathematical programming formulation is developed in Section 3. Section 4 proposes the heuristic algorithm for solving the CVRP. Computational results on Solomon instances are reported in Section 5. Finally, conclusions and future work are presented in Section 6.

2 LITERATURE REVIEW

Early, constructive heuristics, such as saving method (Clarke and Wright, 1964), sweeping method (Gillet and Miller,1974), and Mole and Jameson heuristic (Mole and Jameson,1976) were popular for CVRP. In general, they provide solutions at 10-20% above the optimum, in negligible running times.

Tabu search that constituted the most competing algorithms in the 1990s is still present via variants that include sophisticated memory mechanisms. In 1996, Glover [5] presented the advances, applications, and challenges in tabu search and adaptive memory programming. The main idea is to extract a sequence of points (called bones) from a set of solutions and gen-

*Corresponding author: zh_jiashan@163.com

erate a route using adaptive memory. Further, the adaptive large neighborhood search (ALNLS) [6] is presented by Pisinger and Ropke (2007), in which a control layer adaptively chooses among a number of removal and insertion heuristics to intensify and diversify the search. However, the quality of tabu search depends on the quality of initial solution.

Evolutionary algorithms are proved efficient for the CVRP. [7] presents the first hybrid GA for the VRP, which becomes an effective algorithm available for the large-scale instances generated by Golden et al. [8] presents an improved genetic algorithm for solving the largest existing benchmark instances of CVRP. [9] presents an Parallel Simulated Annealing for large-scale instances. However, the EA is slower than many TS algorithms.

[10] presents an efficient variable neighborhood search heuristic for the capacitated vehicle routing problem. A strategy of the guided local search metaheuristic is used to help escape local minima.

Proposed by Fisher and Jaikumar [11], Cluster First-route Second method, is an effective way to deal with CVRP, especially large scale CVRP. The approach, first creates customer clusters, and then optimizes the order of visits for each cluster as a TSP sub-problem. In [12], the customers were firstly divided into districts according to the main road grid system. Then the customer districts were assigned to vehicles using the vehicle flow formulation model. Experiments show that the method can found high-quality solutions. In the method, clustering efficiently is the key of solving problems.

3 PROBLEM DESCRIPTION AND FORMULATION

Let $G=(V,E)$ be a complete undirected graph with |V|=n+1 nodes. The node $v_0 \in V$ represents a depot, where a fleet of identical vehicles is based, and where the product to be distributed is stored. The other nodes $v_i \in V \backslash \{v_0\}$, for $i \in \{1, ..., n\}$, represent the customers, characterized by demands for non-negative amounts of product q_i. Edges $\{i,j\} \in E$ represent the possibility of traveling directly from a node (customer or depot) $v_i \in V$ to a different node $v_j \in V$ for a transportation cost of c_{ij}. The CVRP aims to find m or less vehicle routes, that is, sequences of deliveries to customers, to visit each customer one time exactly while minimizing the total travel distance. The sum of demands should not exceed on any route of value Q assimilated to the vehicle capacity.

The decision variables of the model are:

$$x_{ij}^{k}=\begin{cases}1, & \text{if customer j is supplied after customer}\\ & \text{i by a vehicle of type k}\\ 0, & otherwise\end{cases}$$

$$y_{jk}=\begin{cases}1, & if \text{ vehicle k visits client j}\\ 0, & else\end{cases}$$

The cost of a vehicle of type k traversing the pair (i, j) is denoted by c_{ij}^{k}. The objective function can be written as follows:

$$\min\sum_{k=1}^{m}f_k\sum_{j=1}^{n}x_{oj}^{k}+\sum_{i=0}^{n}\sum_{j}^{n}\sum_{k=1}^{m}c_{ij}^{k}x_{ij}^{k}$$

Subjected to

$$\sum_{j=1}^{n}x_{oj}^{k}=1, \quad k=1,2,\ldots m \qquad (1)$$

$$\sum_{i=1}^{n}x_{ip}^{k}-\sum_{j=0}^{n}x_{pj}^{k}=0, \quad p=0,1\ldots n \qquad (2)$$

$$\sum_{i=1}^{n}q_i y_i^{k}\le Q, \quad k=1,2,\ldots m \qquad (3)$$

$$\sum_{k=1}^{m}y_{ik}=\begin{cases}1, & i=1,2,...n\\ m, & i=0\end{cases} \qquad (4)$$

$$\sum_{j=0}^{n}x_{ijk}=y_{ik}, \quad i=1,2,\ldots n \qquad (5)$$

$$x_{ij}^{k}\in\{0,1\}, \forall i,j\in V; k=1,2,\ldots m$$

Constraints (1) and (2) state that each vehicle leaves the depot, after arriving at a customer, the vehicle leaves again, and finally returns to the depot. Constraint (3) guarantees that the vehicle capacity will not be exceeded. Constraint (4) and (5) ensure that each client's demand is fulfilled by exactly one vehicle.

3. Improved Ant-Clustering Algorithm

The improved Ant-Clustering Algorithm is based on [13]. Initially, the nodes are scattered randomly on a discrete 2D board. The board can be considered as a matrix of m×m cells, where m^2= 4n, and n is the total number of nodes to be clustered.

At first, K-means is used to cluster the objects (nodes) to form heaps, which include 2 or more objects in a single cell. Given heap H with n_H objects, the parameters are defined as follows:

The maximum distance between two objects in the heap:

$$d_{\max}(H)=\max_{o_i,o_j\in H}d(o_i,o_j)$$

Where $d(o_i,o_j)$ is euclidian distance.

The center of heap:

$$o_{cen}(H)=\frac{1}{n_H}\sum_{o_i\in H}o_i$$

The average distance between the objects and the center of heap:

$$d_{avg}(H) = \frac{1}{n_H} \sum_{o_i \in H} d(o_i, o_{cen}(H))$$

$o_{dissimiar}(H)$ is the object farthest from the center of heap.

At second, Ant-Clustering Algorithm is used as follows:

The ants are randomly scattered throughout the board. $d(o_i, o_j)$ measures the similarity between the pair of elements (o_i, o_j).

$$d(o_i, o_j) = \begin{cases} 0, & o_i \text{ is similar with } o_j \\ 1, & otherwise \end{cases} \quad (6)$$

The normalizing term s^2 equals the total number of sites in the local area, and introduces similarity density function $f(o_i)$, which is a local estimation of the density of nodes and their similarity to o_i.

$$f(o_i) = \begin{cases} \frac{1}{s^2} \sum_{o_j} [1 - \frac{d(o_i, o_j)}{\alpha}], & if \ f > 0 \\ 0, & otherwise \end{cases} \quad (7)$$

Where the constant α scales the similarities. If α is too large, it becomes difficult that ants pick up the object, but it is easy to put down. Thus, objects, which are dissimilar, are clustered in the same cluster. Otherwise, α is too small, objects, which are similar, are not clustered in the same cluster. In this paper, the adaptive strategy is adopted to determine the value of α to improve the clustering Quality. It is shown as follows.

The initial value of α is set to 0.1. During the continuous n times iteration, times of ant's failure to lay down the object is N_f, and the ratio of N_f and N is k. α is given by

$$\alpha = \begin{cases} \alpha + 0.01, & k \geq 0.99 \\ \alpha - 0.01, & k < 0.99 \end{cases} \quad (8)$$

In the process of clustering, parameters are maintained adaptive changes, which make clustering process more robust.

When the ant is not carrying any objects, it looks for possible objects to pick up by looking at the eight neighboring cells around its current position. The algorithm for picking up is shown as follows.

(1) Label the 8 neighboring cells around the ant as "unexplored"

(2) Repeat.

Consider the next unexplored cell around the ant. If the cell is empty, then the object carried by the ant, o_i is dropped with a probability. If the cell is not empty, and the cell contains a single object o_i, then the object o_i is picked up with probability,

$$P_p(o_i) = (\frac{k_1}{k_1 + f(o_i)})^2,$$

Where k_1 and k_2 are threshold constants.

If the cell contains a heap of two objects, then the heap is destroyed by picking up a random object with a probability $P_{destr.}$

If the cell contains a heap H of more than 2 objects, then the most dissimilar object of H is removed only if

$$\frac{D(o_{dissimiar}(H), o_{cen}(H))}{D_{avg}(H)} > p_{remove}.$$

Label the cell as "explored".

(3) Until all the neighboring cells have been explored.

When the ant is carrying an object, then it examines the 8 cells surrounding its current location. The algorithm for dropping the object is shown as follows.

(1) Label the 8 neighboring cells around the ant as "unexplored"

(2) Repeat.

Consider the next unexplored cell around the ant.

If the cell is empty then the object carried by the ant, o_i is dropped with a probability,

$$P_d(o_i) = \begin{cases} 2f(o_i), & if \ f(o_i) < k_2 \\ 1, & if \ f(o_i) \geq k_2 \end{cases}.$$

If the cell contains a single object o_j, then a heap of two objects is created by dropping o_i on o_j only if

$$\frac{D(o_i, o_j)}{D_{\max}} < P_{create}.$$

If the cell contains a heap H then o_i is dropped on H only if

$$D(o_i, O_{cen}(H)) < D(o_{dissimiar}(H), o_{cen}(H)).$$

Label the cell as "explored"

(3) Until all the neighboring cells have been explored.

Meanwhile, the total load of cluster cannot exceed the vehicle capacity. Otherwise, the nodes will merge in another cluster.

4 ALGORITHM FLOW

Phase I is intended to reduce the computational burden of the subsequent solution phase. By establishing the mathematical model in terms of a few clusters rather than a huge number of customers, the CVRP size can be decreased evidently.

Phase II and III are not different from other two–phase method, which is shown in [14]. That is, Phase II aims to assign clusters to vehicles and then sequence them on the same tour by solving a compact version of CVRP. Replacing the customer nodes, the clusters generated in Phase I make problem size shrink sharply. Phase III orders nodes within clusters for every tour, and the nodes will be visited in sequence. To reach the goal, a low-size CVRP will be solved as many times as the number of tours found in Phase II. Actually, each single-tour scheduling problem is tackled as TSP, which includes the same nodes in the clusters.

The main ant-based clustering algorithm is presented as follows.

1) Initially, randomly place the ants on the board. Randomly place nodes on the board at most one per cell.

2) K-means is used to cluster the objects (nodes) to form heaps.

3) Repeat.

For each ant Do

(a) Move the ant;

(b) If the ant does not carry any object then:

If there is an object in the 8 neighboring cells of the ant, the ant possibly picks up the object;

Else,

The ant possibly drops a carried object, by looking at the 8 neighboring cells around it.

End if,

updated α according to (8);Until stopping criteria.

4) Assign clusters to vehicles and then sequence them on the same tour.

5) Genetic algorithm (GA) [15] is implemented within each cluster to order nodes.

5 EXPERIMENTS

In this section, we report our computational results and compare them with those from the existing literature. The proposed algorithm has been executed on an Intel Pentium 4 machine with 2GB of memory, running windows. Our computational experiment is based on the Solomon instances developed by Solomon et al. (1987).

Table 1 (a) Clusters generated for example C-105(50 nodes) in the first phase

	Assigned nodes	Cluster load	Travel distance	Loading rate
C1	14 18 19 20 16 17 15 13	190	95.8847	95%
C2	33 34 35 37 40 39 38 36 32	200	97.9947	100%
C3	48 50 51 49 46 45 41 42 43 44 47	140	65.2956	70%
C4	2 3 5 4 6 8 7 10 9 12 11	160	59.6846	80%
C5	21 22 23 25 26 28 30 31 29 27 24	170	51.2111	85%

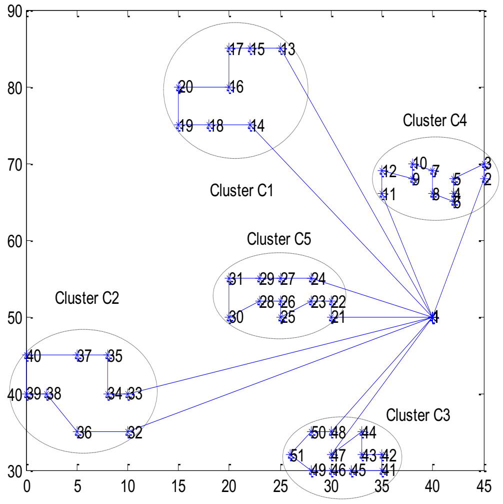
Example C-105(50), R-101(50) and RC-104 (50) have been derived from the original Solomon problem C-105, R-101(50) and RC-104 by just considering the first 50 customers. In the first phase, the clustering procedure is applied to problem C-105(50), R-101(50) and RC-104(50). The 50 original nodes have been merged into customer clusters in a very short time. The nodes in each cluster are shown in Table 1.

Table 1 (b) Clusters generated for example R-101(50 nodes) in the first phase

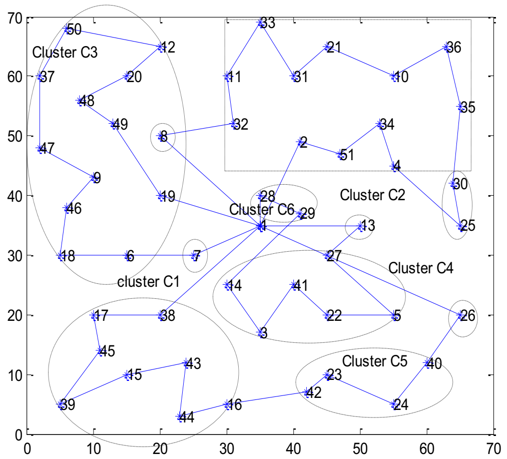
	Assigned nodes	Cluster load	Travel distance	Loading rate
C1	38 17 45 39 15 43 44 16	101	81.6626	50.5%
C2	2 51 34 4 25 30 35 36 10 21 31 33 11 32	183	132.6438	91.5%
C3	19 49 48 20 12 50 37 47 9 46 18 6	193	115.7707	96.5%
C4	27 5 22 41 3 14	86	65.8926	43%
C5	42 23 24 40	83	42.7068	41.5%
C6	29 28	32	13.4164	16%
Discrete nodes	8 7 26 13	5,3,6,19	----	------

Table 1 (c) Clusters generated for example RC-104(50 nodes) in the first phase

	Assigned nodes	Cluster load	Travel distance	Loading rate
C1	2 4 6 46 5 47 9 8 7 3	170	37.4403	85%
C2	11 12 13 15 48 18 17 16 14 10	200	37.8310	100%
C3	25 23 21 50 20 19 49 22 24 26	160	33.8255	80%
C4	43 45 44 41 37 36 38 39 40 42	150	37.4272	75%
C5	35 33 31 29 32 30 28 27 34	170	49.9686	85%
Discrete nodes	51 (merged in C5)	30	---	15%



(a)

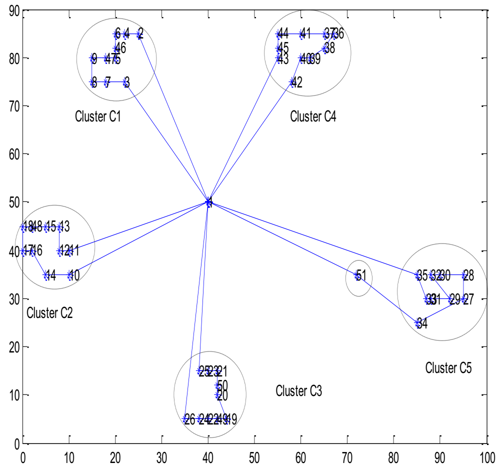


(b)

The improved ant-clustering algorithm has a good performance in clustering nodes, as denoted in Table 1 (a-c). It makes full preparations for Phase II and Phase III. Figure 1 (a-c) shows the truly problem optimum is already found through the exact approach. The method is very successful for clustered instances, such as C-class problems, and the optimum for many of them is retained. See Figure1 (a). It is also effective for RC-class problems, which is shown in Figure 1 (c), but not succeed in R-class problems. See Figure 1 (b).

6 CONCLUSION AND FUTURE WORK

This paper introduces a new ant-clustering algorithm for the capacitated Vehicle Routing Problem. The method aims to integrate a heuristic clustering algorithm into an optimization framework. The introduction of preprocessing phase to gather nodes into a few clusters makes the CVRP size decreased sharply. The proposed method can retain optimum in a reasonable time, especially doing well in solving large-scale CVRP with more than 100 nodes. It is robust as the optimization method. Experiments show that ant-clustering algorithm can succeed in solving a variety of Solomon problems.



(c)

Figure 1. Best solution found for problem

Real-life vehicle routing application is more complicated. For example, the requirement of customers is often uncertain. The extension of the method to these more difficult problems is worth further research.

ACKNOWLEDGEMENT

This paper is Supported by the National Natural Science Foundation of China (No. 50904032) and the Education Department of Liaoning province science and technology research project (No.L2010177)

REFERENCES

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

[2] Christofides, N. 1976. The vehicle routing problem. *RAIRO Operations Research*, 10 (2): 55-70.

[3] PaoloT. & Daniele V. 2002. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123: 487-512.

[4] Lenstra, J. K., & Rinnoy Kan, A. H. G. 1981. Complexity of Vehicle and Scheduling Problems. *Networks*, 11: 221-227.

[5] F. Glover. 1996. Tabu search and adaptive memory programming-Advances, applications, and challenges.

Interfaces in Computer Science and Operations Research.

[6] Pisinger D, Ropke S. 2007 A general heuristic forvehicle routing problems. *Computers & Operations Research*, 34: 2403-2435.

[7] C. Prins, 2004. A simple and effective evolutionary algorithm for the vehicle routing problem, *Computers & Operations Research*, 31 (12): 1985-2002.

[8] B. Dorronsoro, D. Arias. 2007. A Grid-based hybrid cellular genetic algorithm for very large instances of the VRP. *Parallel and Grid Computing for Optimization*, PGCO.

[9] Czech Z. J, Czarnas P. Parallel. 2002. Simulated Annealing for the Vehicle Routing Problem with Time Windows. *Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*, pp: 376-379

[10] J. Kytojoki, & T. Nuortio et al. 2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems, *Computers & Operations Research*, 34(9): 2743-2757.

[11] Fisher, M. & Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. *Networks*, 11(2): 109-124.

[12] Z. W. Qu, L. & N. Cai et al. 2004. Solution framework for the large scale vehicle de-liver/collection problem, *Journal of Tsinghua University (Sci. & Tech.)*, 44(5): 581-584.

[13] J.L. Deneubourg, S. Goss, et al. 1991. The dynamics of collective sorting robot-like ants and ant-like robots, In Proc. of the 1st Conf. on Sim of Adaptive Behavior.

[14] Christian PRINS. 2008. The route-first cluster-second principle in vehicle routing, *Oslo*, 06.

[15] Barrie M. Baker, M. & A. Ayechew. 2003. A genetic algorithm for the vehicle routing problem, *Computers & Operations Research*, 30(5): 787-800.