Research of Multi-Depot Vehicle Routing Problem by Cellular Ant Algorithm

Yuanzhi Wang
Geography Department, Dezhou University, Dezhou, Shandong 253023, China
Email: wangyuanzhi1@sina.com

***Abstract*—The Multi-Depot Vehicle Routing Problem (MDVRP) is a generalization of SDVRP, in which multiple vehicles start from multiple depots and return to their original depots at the end of their assigned tours. The MDVRP is NP-hard, therefore, the development of heuristic algorithms for this problem class is of primary interest. This paper solves Multi-Depot Vehicle Routing Problem with Cellular Ant Algorithm which is a new optimization method for solving real problems by using both the evolutionary rule of cellular, graph theory and the characteristics of ant colony optimization. The simulation experiment shows that the Cellular Ant Algorithm is feasible and effective for the MDVRP. The clarity and simplicity of the Cellular Ant Algorithm is greatly enhanced to ant colony optimization.**

***Index Terms*—Multi-Depot Vehicle Routing Problem (MDVRP), Cellular Ant Algorithm, Graph theory**

I. Introduction of MDVRP

The Vehicle Routing Problem (VRP) has been one of the central topics in optimization since Dantzig proposed the problem in 1959 [1]. The Vehicle Routing Problem involves the design of a set of minimum-cost vehicle routes, originating and terminating at depot, for a fleet of vehicles that services a set of customers with known demands. Each customer is serviced exactly once and, furthermore, all customers must be assigned to vehicles without exceeding vehicle capacities [2] [3].

In the Single-Depot Vehicle Routing Problem (SDVRP), multiple vehicles leave from a single location and must return to that location after completing their assigned tours. The Multi-Depot Vehicle Routing Problem (MDVRP) is a generalization of SDVRP, in which multiple vehicles start from multiple depots and return to their original depots at the end of their assigned tours.

The MDVRP is NP-hard [4] [5], therefore, the development of heuristic algorithms for this problem class is of primary interest.

A. The Description of MDVRP on Graph

Definition 1 (graph) In the most common sense of the term, a graph is an ordered pair G = (V, E), comprising a set V of vertices or nodes together with a set E of edges or lines, which are 2-element subsets of V.

Definition 2 (undirected graph) An undirected graph is one in which edges have no orientation. The edge (a, b) is identical to the edge (b, a), i.e., they are not ordered pairs, but sets {u, v} (or 2-multisets) of vertices.

Definition 3 (degree) In graph theory, the degree of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice. The degree of a vertex *v* is denoted deg(*v*).

Definition 4 (regular graph) In graph theory, a regular graph is a graph where each vertex has the same number of neighbors; i.e. every vertex has the same degree.

Definition 5 (complete graph) In the mathematical field of graph theory, a complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge. The complete graph on n vertices has n(n − 1)/2 edges (a triangular number). It is a regular graph of degree n − 1.

MDVRP is defined by complete graph G = (V, E). V is a set of vertices. Vertex set V is divided into two subsets $V_c = (v_1, v_2, ..., v_n)$ and $V_d = (v_{n+1}, v_{n+2}, ..., v_{n+M})$. V_c is a set of vertices composed of customers and V_d is a set of vertices composed of depots. For each customer node, there is a negative demand q_i. For each depot node, there are $K_i (i = 1,..., M)$ vehicles. Each vehicle k has its load w. $E = (V_i, V_j)$ is a set of edges. For each edge (V_i, V_j) has a nonnegative cost c_{ij} which indicates the cost from customer i to customer j.

Note: For all the i, j, $c_{ij} = c_{ji}$; For all the i, j, k, $c_{ik} \leq= c_{ij} + c_{jk}$.

B. The model of MDVRP

First, the following variable is defined:

$$x_{ijmk} = \begin{cases} 1 & \text{, If the vehicle k of the depot m runs from customer i to customer j} \\ 0 & \text{, otherwise} \end{cases} \quad (1)$$

The objective function is

$$\min Z = \sum_{i=1}^{n+M} \sum_{j=1}^{n+M} \sum_{m=1}^{m} \sum_{k=1}^{K_m} c_{ij} x_{ijmk} \quad (2)$$

Project supported by Shandong province office of education (No.S07WZ35)
Corresponding author. wangyuanzhi1@sina.com (Yuanzhi Wang).

doi:10.4304/jcp.8.7.1722-1727

s.t.

$$\sum_{j-1}^{n}\sum_{k=1}^{K_m} x_{ijmk} \le K_m, \quad i = m \in \{n+1, n+2, ..., n+M\} \tag{3}$$

$$\sum_{j=1}^{n} x_{ijmk} = \sum_{j=1}^{n} x_{jimk} \le 1, i = m \in \{n+1, n+2, ..., n+M\}, k \in \{1,2,...,K_m\} \tag{4}$$

$$\sum_{j=1}^{n+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ijmk} = 1, i \in \{1,2,...,n\} \tag{5}$$

$$\sum_{i=1}^{n+M}\sum_{m=1}^{M}\sum_{k=1}^{K_m} x_{ijmk} = 1, j \in \{1,2,...,n\} \tag{6}$$

$$\sum_{i=1}^{n} q_i \sum_{j=1}^{n+M} x_{ijmk} \le w, m \in \{n+1, n+2, ..., n+M\}, k \in \{1,2,...,K_m\} \tag{7}$$

$$\sum_{j=n+1}^{n+M} x_{ijmk} = \sum_{j=n+1}^{n+M} x_{jimk} = 0, i = m \in \{n+1, n+2, ..., n+M\}, k \in \{1,2,...,K_m\} \tag{8}$$

In equation (2), c_{ij} represents the cost from customer i to customer j, which usually mean the distance from customer i to customer j. m is the number of the depot and k is the number of the vehicle. Equation (3) guarantees that the number of vehicles starting from the depot cannot exceed the vehicles owned by the depot, in which K_m represents the total vehicles of the m depot. Equation (4) ensures that the vehicles start from their respective depot, and return to the same depot. Equation (5), (6) guarantees that each customer is only visited by one vehicle only one time, but it is allowed that other vehicles pass the customer, and there is no limit of the pass times. Equation (7) ensures that the total demand of each customer is no more than the capacity of one vehicle on each distribution lines. Equation (8) guarantees that vehicles can't directly run from one depot to another depot.

II. The Improvement of Cellular Automaton for the Cellular Ant Algorithm

A. Introduction of Cellular Automaton

A cellular automaton (abbrev.CA) is a discrete model studied in computability theory, mathematics, physics, complexity science, theoretical biology and microstructure modeling. It consists of a regular grid of cells, which are in one of a finite number of states, such as "On" and "Off" (in contrast to a coupled map lattice). The grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood (usually including the cell itself) is defined relative to the other cell. An initial state (time t=0) is selected by assigning a state to each cell. A new generation is created (advancing t by 1), according to some fixed rule (generally, a mathematical function) that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. Typically, the rule for updating the state of cells is the same for each cell and does not change over time, and is applied to the whole grid simultaneously, though the exceptions are known.

A CA consists of four components: cell, state, cellular space, neighborhood.

Definition 6 (cell). Cell is the basic components of CA. Cell distributes in one dimensional, two dimensional or multidimensional discrete Euclidean lattice.

Definition 7 (cellular space). It is a lattice of N identical finite-state machines (cells), each with an identical pattern of local connections to other cells for input and output, along with boundary conditions if the lattice is finite.

Definition 8 (neighborhood). For each cell, a set of cells is called its neighborhood (usually including the cell itself) defined by the specified rule.

Definition 9 (rule). Rule(transition function) gives the update state for each cell.

An cellular automaton is represented formally by a 4-tuple $A = (L, S, N, f)$, where

L is cellular spaces;

S is a finite set of states;

$N = (s_1, s_2, ..., s_n)$, n is the neighbor number of center cell;

s_i represents a state of cell;

f is the transition function, that is, f: $s_n \rightarrow S$.

In order to use cellular automata in solving MDVRP, it is needed to combine CA and graph theory.

B. The Combination of Cellular Automaton and Graph

Firstly, all the vertices in graph will be defined as cells which are the basic components of CA. So cellular spaces in graph is

$$L = \{ v_1, v_2, ..., v_n \}.$$

$\{ v_1, v_2, ..., v_n \}$ is the set of vertices of G = (V, E).

In CA theory, the von Neumann neighborhood and the Moore neighborhood are the two most commonly used neighborhood types.

In cellular automata, the von Neumann neighborhood comprises the four cells orthogonally surrounding a central cell on a two-dimensional square lattice. The neighborhood is named after John von Neumann, who used it for his pioneering cellular automata including the Universal Constructor.

The Moore neighborhood comprises the eight cells surrounding a central cell C on a two-dimensional square lattice. The neighborhood is named after Edward F. Moore, a pioneer of cellular automata theory.

Of course, there are other neighborhood types, but the basic principle of these types is that cells can influence each other, whose distance between each other is short. The principle can be called the close principle. At certain times, the close principle is right to biological cells in the body. But in some cases, for two cells which are farther together, the interaction between them are relatively large.

The close principle is not fit for the cellular automaton on graph. In graph, obviously to two vertices which are close together, if there is no edge directly connected, the interaction between them is weak. Also, to two vertices which are farther together, if there is an edge connecting them directly, the interaction between them is relatively great. For example, in Figure 1, vertex 1 is closer to vertex 5 than vertex 2, but there is no edge directly connecting vertex 1 and vertex 5, so the interaction between vertex 1 and vertex 5 is weaker than the interaction between vertex 1 and vertex 2. The same relation lies in vertex 1, vertex 2 and vertex 5; vertex 1, vertex 4 and vertex 5; vertex 3, vertex 2 and vertex 6; vertex 3, vertex 4 and vertex 6.

Finally, in graph, the definition of central cell's neighborhood is

$$N(v_x) = \{v | (v_x, v) \in E \vee (v, v_x) \in E \vee v = v_x\}.$$

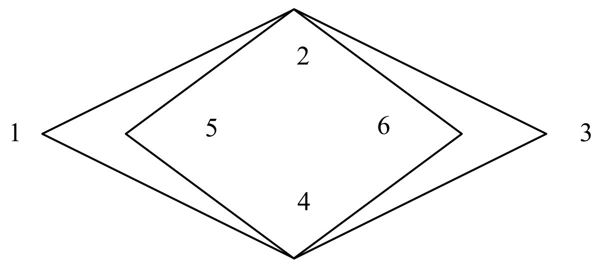

Figure 1. The cellular neighborhood on graph

III. THE MODEL OF ACO ON GRAPH

The framework of ACO is inspired by the observation, made by ethologists, that ants use pheromone trails to communicate information regarding the shortest paths to food [6]. The ACO metaheuristic, formally developed by Dorigo, Di Caro, and Gambardella draws its inspiration from the experimental observations of emergent behavior in real ant colonies, such as the research and experimentation of Goss et al. on a laboratory-contained colony of Argentine ants [7][8].

Gambardella and Dorigo (1996) have experimented with ACS for solving both symmetric and asymmetric instances of the TSP [9]. Gambardella, Taillard, and Agazzi utilized multiple, cooperative ACS colonies, each designed to optimize a specific objective function, for solving instances of the VRPTW. Results across 56 VRPTW benchmark problems were compared with those of several other related studies and were found to be equally competitive with the best-known solutions [10].

Donati et al. has more recently applied the ACS algorithm for solving the TDVRP, in which the researchers developed the concept of time-dependent pheromones by partitioning time into periods during which overall network speeds can be considered constant and associating differing sets of pheromones to each of these sub-space "time slices". Results showed that this time-dependent approach proves quite valuable when applied to VRPs with dynamically changing network conditions [11].

Other highly relevant ACO research (besides that of ACS) applied to the field of VRP includes the work of Bullnheimer, Hartl, and Strauss. The research of this group focuses on the heavily related Ant System (AS) algorithm for solving the VRP. One of the more significant contributions made by this group towards using ACO algorithms to solve VRPs is their use of candidate lists [12].

If ant colony is put on graph G = (V, E), For ant k, the probability P_{ij}^k of moving from vertex i to vertex j depends on the combination of two values, viz., the attractiveness η_{ij} of the move, as computed by some heuristic indicating the a priori desirability of that move and the trail level τ_{ij} of the move, indicating how proficient it has been in the past to make that particular move.

In general, the k'th ant moves from vertex i to vertex j with probability

$$P_{ij}^k = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum (\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})},$$

Where ,

τ_{ij} is the amount of pheromone deposited for transition from cell i to j,

$0 \le \alpha$ is a parameter to control the influence of τ_{ij} ,

η_{ij} is the desirability of vertex transition ij (a priori knowledge, typically 1 / c_{ij}, where c_{ij} is the transportation cost between vertex i and vertex j),

$\beta \ge 1$ is a parameter to control the influence of η_{ij} .

Pheromone update is as follows:

When all the ants have completed a solution, the trails are updated by

$$\tau_{ij}^k = (1-\rho)\tau_{ij}^k + \Delta\tau_{ij}^k,$$

Where,

τ_{ij}^k is the amount of pheromone deposited for a transition i to j,

$\tau_{ij}(0)$ is the initial pheromone level assumed to be a small positive constant distributed equally on all the paths of the network since the start of the survey, ρ is the pheromone evaporation coefficient,

$\Delta\tau_{ij}^{k}$ is the amount of pheromone deposited,

$$\Delta\tau_{ij}^{k} = \begin{cases} Q/L_k & \text{if ant k uses curve ij in its tour} \\ 0 & \text{otherwise} \end{cases}$$

where L_k is the cost of the k'th ant's tour (typically length) and Q is a constant.

IV. The Cellular Ant Algorithm for MDVRP

The Cellular Ant Algorithm combines ant colony optimization with cellular automaton to refrain from falling in to partial optimization while the good optimization ability of ant colony optimization is reserved.

ZhuGang first completely proposes Cellular Ant Algorithm for function and discrete systems optimization based on ant algorithm and cellular automata in his Ph.D. thesis. His research provides a new kind algorithm for NP-hard problems and gives convergence proof of the algorithm. His paper solves the classical TSP by Cellular Ant Algorithm through series of typical instances. The computational results show the effectiveness of the algorithm in numerical simulation [13].

The above algorithm is not designed for MDVRP. The Cellular Ant Algorithm for MDVRP is following.

The thought solving MDVRP with Cellular Ant Algorithm is to convert multi-depot problem into single depot problem. First, assume that there is a virtual depot, and each customer or actual depots are all customers of the virtual depot. Then all distribution vehicles set out from the virtual depot to each customer.

Cellular automata $A = (L, S, N, f)$ is constructed on n vertex weighted graph G (V, E), where, cellular space is

$$L = \{v_0, v_1, ..., v_n, v_{n+1}, v_{n+2}, ... v_{n+M}\};$$

v_0 represents the virtual depot. $v_1, v_2 ... v_n$ represent the customers and $v_{n+1}, v_{n+2}, ... v_{n+M}$ represent the depots. State set is S = (S_N, S_M) where S_N means that the vertex is in initial state, and is not searched by ant; S_M indicates that the vertex is in mature state, that is the road has been searched by ant.

But there are four significant differences between SDVRP and MDVRP that has a virtual depot,

(1) The cost (including time, journey, fuel consumption, etc.) from the virtual depot to actual depots is zero. This point is guaranteed by the following expressions,

$$c_{ij} = c_{ji} = 0, i = 0, j = n+1, n+2, ... n+M .$$

(2) The vehicle starting from the virtual depot can only be to the actual depots, then finish distribution task according to the requirements of the customers. This point is guaranteed by the following expressions,

$$P_{ij}^{k} = 1, i = 0, j = n+1, n+2, ... n+M .$$

(3) The vehicle finishing distribution task can go back to the virtual depot, but its terminal point is recorded as the actual depot which the vehicle starts from.

(4) The vehicle can not service the other actual depot after it has serviced one actual depot. This point is guaranteed by the following expressions,

$$P_{ij}^{k} = 0, i = 1, 2, ..., n, j = n+1, n+2, ... n+M .$$

The overall procedure of the Cellular Ant Algorithm for MDVRP is following.

Step1: initialization:

t=0, $\tau_{ij}(0) = \text{const}, \Delta\tau_{ij} = 0$, The m ants is located in the virtual depot, NC=0, $w_k = 0$, $CA_{(\min)} = 0$;

Step2: let s=1

for (k=1; k<=m; k++)

After k ant starts from the virtual depot to the first visiting cellular (the actual depot) according to probability P_{ij}^{k}, the state of the first visiting cellular is to S-M and the cause of S-M is remembered as k ant;

Step3: while (the state of all cellulars is not S-M)

{

s=s+1;

for (k=1; k<=m; k++)

{

According to probability P_{ij}^{k}, the next destination (cellular or customer) is chosen. The k ant is moved to cellular j. Next the state of the cellular j is to S-M and the cause of S-M is remembered as k ant. If cellular j is the other actual depot, the probability $P_{ij}^{k} = 0$ .If cellular j is the virtual depot, the finding route activity of the ant is stopped.

}

}

Step4: for(k=1; k<=m; k++)

if (the ant k returns to the virtual depot)

{

The total path cost W_k of k ant is calculated; the minimum cost routes found are updated; the cost $CA_{\min}$ which corresponds to the minimum cost route is recorded.

}

For (k=1; k<=m; k++)

The pheromone is updated;

Step5: calculates all the $\tau_{ij}(t+1)$

t=t+1;

NC=NC+1;

$\Delta\tau_{ij} = 0$;

Step6: if (NC<$NC_{(MAX)}$)

{

all cellulars convert to the state of S _N;

back to step1;

}

else back to step5

Step7: The minimum cost routes found are updated; the cost $CG_{\min}$ which corresponds to the minimum cost route is recorded.

If (| $CG_{(\min)}$ - $CA_{(\min)}$ | < Given any small positive Numbers)

The minimum cost routes are found and the whole program is terminated;

Else

Back to step2

V. The Simulation Experiment

The experimental data is following: coordinates and demand of 15 customers are shown in table 1. Vehicle number and coordinates of depots are shown in table 2.

It is assumed that coordinates of the virtual depot are (50, 50). The Cellular Ant Algorithm for MDVRP is coded in C++ and executed 20 times on a PC. The 20 results are obtained: 493.64, 488.01, 490.07, 489.35, 489.94, 489.46, 490.90, 489.04, 489.42, 493.19, 490.86, 484.23, 487.96, 492.94, 494.43, 490.93, 492.12, 492.04, 492.50, 492.52. So the shortest distance, that is, the least cost is 484.23. The specific situation of the route is shown in table 3.

TABLE I. Number, coordinates and demand of customers

customer	1	2	3	4	5	6	7	8
X coordinate	19	33	73	49	70	27	10	39
Y coordinate	0	3	85	73	94	44	69	25
demand	1.0	1.8	1.1	0.6	1.9	1.4	1.2	0.2
customer	9	10	11	12	13	14	15	
X coordinate	16	68	10	83	88	32	70	
Y coordinate	81	76	57	43	52	58	18	
demand	1.7	0.8	0.9	0.8	1.9	1.6	0.9	

TABLE II. Vehicle number and coordinates of depots

depot	A	B	C
X coordinate	33	26	57
Y coordinate	77	30	0
Vehicle number	3	2	1

TABLE III. distribution data with Cellular Ant Algorithm

origin depot	distribution route	weight distribution	distribution mileage	Total distribution mileage
A	A→9→7→11→A	3. 8	73. 76	484.23
A	A→5→3→10→A	3. 8	95. 52	
A	A→4→13→12→A	3. 3	131. 54	
B	B→8→2→1→B	3. 0	81. 86	
B	B→6→14→B	3. 0	57. 55	
C	C→15→C	0. 9	44. 40	

VI. Conclusions

MDVRP is a typical NP hard problem. The above results lack of the optimal solution based on Method of Exhaustion to serve as reference, so the optimal solution of the paper is uncertain to be the global optimal solution. From the experiment process, calculation time that is executed on personal computer in each experiment is not more than 10 seconds. The short calculation time shows that the Cellular Ant Algorithm has a higher efficiency. The clarity and simplicity of the Cellular Ant Algorithm is of better performance than other methods. At least, it can be said that the optimal solution in the paper is global satisfied solution. The Cellular Ant Algorithm combines cellular automaton and ant colony optimization which cooperating together to optimize Multi-Depot Vehicle Routing Problem. The above experiment shows that the Cellular Ant Algorithm is feasible and effective for the MDVRP.

References

[1] G.B. Dantzig, J.H. Ramser, "The Truck Dispatching Problem", *Management Science*, 1959.

[2] Chunyu Ren, "Research on Single and Mixed Fleet Strategy for Open Vehicle Routing Problem", *Journal of Software*, VOL. 6, NO. 10, 2011, pp. 2076-2081.

[3] Jiangqing Wang, Rongbo Zhu, "Efficient Intelligent Optimized Algorithm for Dynamic Vehicle Routing Problem", *Journal of Software*, VOL. 6, NO. 11, 2011, pp. 2201-2207.

[4] L. BODIN, B. GOLDEN, A. ASSAD and M. BALL, "Routing and scheduling of Vehicles and Crews: The State of the Art", *Computers and Operations Research*, 10, 1983.

[5] J. LENSTRA, A.RINNOOY KAN, "Complexity of Vehicle Routing and Scheduling Problems", *Networks*, 11(1981), pp. 221-228.

[6] Jingyao Li, Shudong Sun, Yuan Huang, "Adaptive Hybrid Ant Colony Optimization for Solving Dual Resource Constrained Job Shop Scheduling Problem", *Journal of Software*, VOL. 6, NO. 4, 2011, pp. 584-594.

[7] Dorigo, M., G. Di Caro, and L.M. Gambardella, "Ant Algorithms for Discrete Optimization", *Artificial Life* 5(2), 1999, pp. 137-172.

[8] Goss, S., et al, Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, 76, 1989, pp. 579-581.

[9] Dorigo, M. and L.M. Gambardella, "Ant colonies for the traveling salesman problem", Universit'e Libre de Bruxelles, Belgium, IRIDIA, *Technical Report TR/IRIDIA/*1996-3.

[10] Gambardella, L.M, E. Taillard, and G. Agazzi, "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows", In: D. Corne, M. Dorigo, and F. Glover (Eds.) *New Ideas in Optimization*, McGraw-Hill, London, UK, 1999, pp. 63-76.

[11] Donati, A.V., et al, "Time Dependent Vehicle Routing Problem with an Ant Colony System", *In: internal IDSIA report*, 2003.

[12] Bullnheimer, B., R.F. Hartl, and C. Strauss , "An improved ant system algorithm for the vehicle routing problem", *Annals of Operations Research*, 89, 1999, pp. 319-328.

[13] ZhuGang. "cellular ant algorithm and its application", *PhD thesis*. University of Shanghai for Science and Technology, 2007.

Yuanzhi Wang, Geography Department, Dezhou University Dezhou, Shandong Province, China, P.C. 253023
Email: wangyuanzhi1@sina.com

Personal Details & Education:
Born Date: October, 1977
Nationality: Hebei Province, China
Major: Computer Science and Application in Geography
Master of Engineering, 2001.9- 2004.7, Henan Polytechnic University, Henan , China
Research Interests & Experience:

- GIS Applications
- Optimization theory and algorithms
- Data Mining
- VRP