THE VEHICLE ROUTING PROBLEMS WITH A LARGE-VOLUME CUSTOMER OR WITH TIME WINDOW CONSTRAINTS BY DECENTRALIZED PROBABILISTIC ALGORITHM

Ichiro Shigaki

Department of Industrial Management
Osaka Institute of Technology,
Omiya 5-16-1, Asahi-ku, Osaka, 535-8585, Japan
E-mail: shigaki@dim.oit.ac.jp

Masami Konishi

Department of Electrical Electronic Engineering
Okayama University,
Tsushima-naka 3-1-1, Okayama, 700-8530, Japan
E-mail: konishi@elec.okayama-u.ac.jp

Abstract

In this paper, we formulated the vehicle routing problem with a large-volume customer or with time constraints using a decentralized probabilistic algorithm, designed using multi-agent system, in which agents autonomously and independently search the partial solution of the problem, their customers and traveling routes, to minimize transportation cost. We present a new formulation based on decentralized probabilistic algorithm which is modified from the ant system. The validity of this method is studied by a computer experiment for the one depot, three agents, one hundred customers, or one depot, thirty customers, three agents with time window constraints.

Keywords: Vehicle routing problem, Traveling salesman problem, Combinatorial optimization problem, Time window constraints

1. INTRODUCTION

We consider the distribution problem in which vehicles based at a central facility (depot) are required to visit geographically dispersed customers in order to fulfill known customer requirements. The problem appears in a large number of practical situations concerning the distribution of commodities and is known by the name: the vehicle routing problem (VRP).

VRP and its origin of the traveling salesman problem (TSP) are typical of other problems of its genre: combinatorial optimization. The increasing hardware power of computers enable us to solve these problems by efficient algorithms like simulated annealing (Arts, et al., 1989), tabu search (Barbarosoglu, et al., 1999), neural networks et al. (1996) proposed, however, a versatile, robust, population based decentralized approach (ant algorithm) to solve TSP and its applied forms to VRP have brought good results against VRP (Kawamura, et al., 1998). We proposed a probabilistic algorithm to VRP (Shigaki, et al., 2001), modified from the ant algorithm with decreasing number of parameters and showed applicability to various VRPs as (1) one depot, thirty customers, and three agents, (2) one depot, thirty customers, and three agents, and different vehicles speed, (3) two depots, thirty customers, and three agents.

This paper describes that the algorithm is useful for the case of one depot, one hundred customers, three agents, and that of one depot, thirty customers with time window constraints.

2. THE MULTI-AGENT VRP

The multi-agent VRP is to route the agents (one route per agent, starting from one or more than one depots and finishing at the same depots), so that all customers are supplied with their demands and the total travel cost (sum of the transportation cost and the loading-unloading cost) is minimized while the differences of costs between agents within a certain value.

In the case of VRP with time window constraints, each customers specifies a time interval within which delivery should take place. The problem is to design a complete tour for each vehicle, starting from and ending at the depot, and servicing all the customers within their time window, at minimum routing cost.

3. FORMULATION OF THE VRP
problem the basic VRP. The basic VRP is given by a set of same vehicles $V$, a set of customers $C$ and a directed network connecting customers is as follows. A customer $c_j (j = 1, \ldots, n)$ has a demand $d_j$. The capacity of vehicle $v_i \in V (i = 1, \ldots, m)$ is $u_i$, and maximum travel cost for vehicle $v_i$ is $c_{fj}$. $f(l_{jk})$ is the cost proportional to the length $l_{jk}$ between customer $c_j$ and customer $c_k$. $g(d_j)$ is the cost for loading and unloading the demand $d_j$. Variables $x_{jk}^i$ are defined, so that the objective function is to be minimized.

$$x_{jk}^i = \begin{cases} 1, & \text{if vehicle } v_i \text{ visits customer } c_k \text{ immediately after customer } c_j, \\ 0, & \text{otherwise}. \end{cases}$$

The VRP is then to minimize the sum of transportation cost and loading-unloading cost,

$$E = \sum_{i} h_i^i$$

where

$$h_i^i = \sum_{j \in C} \sum_{k} (f(l_{jk}) + g(d_j))x_{jk}^i$$

It is easy to see that (1) should be minimized subject to:

$$\sum_{j \in C} \sum_{k} |h_i^i - h_j^i| < a$$

$$\sum_{j \in C} x_{jk}^i = 1 \quad \forall c_j \in C$$

$$\sum_{i,j} x_{jk}^i \leq u_i \quad \forall v_i \in V$$

$$\sum_{j \in C} f(l_{jk}) x_{jk}^i \leq e_i \quad \forall v_i \in V$$

$$h_i^i > h_0 \quad \forall v_i \in V$$

Equation (1) is the objective function of this problem to minimize the sum of $h_i^i$. In equation (2), $h_i^i$ means the sum of transportation cost and loading-unloading cost for agent $i$. Nakano et al. carried out optimization using the following objective function (1)'.

$$\min_{r_j^i \in [0,1]} \left\{ \sum_{i} b_i^i + w \sum_{i,j,k} |h_i^i - h_j^i| \right\}$$

where $w$ is a parameter to balance minimization of cost with equal distribution of cost to agents.

However, preliminary calculations showed that we could obtain better solution by equations (1) and (2) than by equation (1)' in our algorithm. Constraints (3) indicates that the differences of cost between agents should be less than a certain value. Constraint (4) ensures that customer $c_j$ is allocated to some vehicle $v_i$. Constraint (5) denotes that each vehicle has the loading limits. Constraint (6) denotes that each vehicle has the maximum transportation distances. Let minimum of $h_i^i$ be $h_0$, constraint (7) indicates that every $h_i^i$ should exceed $h_0$.

4. PROBABILISTIC ALGORITHM

The proposed search algorithm is based on the multi agent system. It consists of several agents which have the ability to autonomously and independently search the partial solution of the problem.

(a) Initialization

A depot $c_0$ is set at the origin of the coordinates, then, locations of customers are set randomly. Customers are assigned to agents randomly with numbers as same as possible. An initial solution at $t = 0$, then, is composed by the nearest neighbor algorithm. The way to construct a tour is as follows:

1. Start with a partial tour consisting of a depot $c_0$.
2. If the current partial tour is $c_0, \ldots, c_k, k < n/3$, let $c_{k+1}$ be the customer, not currently on the tour, which is closest to $c_k$, and add $c_{k+1}$ to the end of the tour.
3. Halt when the current tour contains all customers and the depot.

One obvious drawback of this algorithm is the fact that, although all earlier edges are in a sense 'minimal,' the final edge $[c_{\text{final}}, c_0]$ may be quite long.

(b) Insertion of a customer to a partial route

Each agent searches new partial solution as follows.
i. Insertion of a customer into a partial solution

An agent $v_i$ is selected randomly, and then a customer $c_k$ for the agent $v_i$ is selected randomly. Let $S'(t)$ be a set of customers assigned to $v_i$. We choose one customer, and let it be $c_h$. The probability of choosing $c_h$ for $v_i$ (viz. transition probability) is calculated by the following equation.

$$Q_i(t) = \frac{\eta_{ih}}{\sum_{j \neq h, \in \eta_{ij}}}$$  \hspace{1cm} (8)

where

$$\eta_{ij} = \frac{1}{l_{ij}}$$  \hspace{1cm} (9)

We call $\eta_{ij}$ visibility, the quantity $1/l_{ij}$, and $l_{ij}$ the length of the route between customers $c_i$ and $c_j$; in this case, $l_{ij}$ is the Euclidean distance (i.e. $l_{ij} = \sqrt{(x_i-x_j)^2 + (y_i-y_j)^2}$). Selection of a customer is proportional to the visibility. Namely, the transition probability increases with decreasing length between customers $c_k$ and $c_h$.

ii. Removal of the customer from the partial solution

If customer $c_h$ is selected for insertion to $S'(t)$, it is removed from the originally belonging route.

iii. Rearrangement of partial solutions

Each tour will be constructed again by the nearest neighbor algorithm for each agents.

iv. Evaluation of routes

Cost is calculated only if constraints (3), (5), (6) are satisfied.

v. Selection of a partial solution

The acceptance probability $R(t)$ of a new solution is defined as

$$R(t) = \begin{cases} 1 & \text{if } \Delta E(t) \leq 0 \\ \exp(-\Delta E(t)/T) & \text{otherwise} \end{cases}$$  \hspace{1cm} (10)

where $E(t)$ is the evaluation function of solution, and $\Delta E(t)$ is the difference between evaluation function of new solution and the last one. Namely, if $\Delta E(t) \leq 0$, we surely select a new solution. But, even if $\Delta E(t) > 0$, we select a new solution at the possibility of $\exp(-\Delta E(t)/T)$. This operation induces to slip out from localized solution. Parameter $T$, constant, controls agent’s policy, therefore suitable value of $T$ should be set to obtain good solutions. In the ant system, four parameters are necessary for calculation such as the relative importance of the trail, the relative importance of the visibility, trail persistence and a constant related to the quantity of trail, which induce us some preliminary calculations. However, in this algorithm, only one parameter is necessary.

(c) Renewal of time

Time is renewed as $t = t + 1$, and the procedure (b), (c) is repeated up to $T_{\text{max}}$.

(d) Output of the result

The whole solution with the lowest cost is output.

5. A COMPUTER EXPERIMENT

A computer experiment is carried out to observe the behaviors and characteristics of agents and examine the efficiency of this algorithm.

5.1 A Solution of Problem 1 (one depot, one hundred customers, three agents)

The first problem consists of 100 customers nodes randomly placed in the plane at $t = 0$ whose apexes are at $(\pm 200, \pm 150)$. The depot is placed in the center of this area. Since we assumed that demands of all customers are unit, each total cost of $h$ for vehicle $v_i$ means a tour length. The capacities of all vehicles are set 38 units. Vehicles are allowed to carry different number of units, but each vehicle should start with more than 33 units.

The minimum cost by this algorithm is 3754 shown in Figure 1. Three routes are well separated, showing similarity to the best solution in the next figure. Contrary to the case of one depot, thirty customers, and three agents, such instances are
Figure 1 Shape of the best solution (one depot, one hundred customers, three agents at t=10,000 by this algorithm).

Figure 2 Shape of the best solution (one depot, one hundred customers, three agents at t=10,000 by the combination of the saving algorithm and this algorithm).

However, few that randomly assigned customers dispersed uniformly in the whole area are divided clearly into 3 geographical customer areas by this calculation.

Then, we tried to combine this algorithm with the sweep algorithm or the saving algorithm. In these methods, we, first, solve the problem by the sweep algorithm or the saving algorithm, then, we carried out calculations by this algorithm.

We used PC with Pentium IV processor of 1.80 GHz and programs are compiled by Borland C++ Builder 6. We could obtain most of the results within several ten seconds.

Figure 2 is the shape of solution by the combination of the saving algorithm and this algorithm at t= 10,000. Three agents have 37, 33, 30 customers, respectively, and the plane is divided approximately into 3 geographical customer areas to serve little overlap each other. The solution, however, is improved by exchanging customers in the boundary areas of three agent’s routes.

We compared solutions calculated by various methods. According to Table 1, we could obtain the best solutions by the combination of the saving algorithm and this algorithm. By this algorithm alone, we could obtain a good solution, but in many times, it is difficult to escape from localized solution. The standard deviation of total cost was 2.1%, and the difference ratio between maximum cost and minimum cost to total cost was 5.1%, showing that we can obtain considerably uniform solutions every time due to the combination of this algorithm to the saving algorithm.

5.2 A Solution of Problem 2 (one depot, thirty customers, three agents with time window constraints)

As far as the geographical and demand data are concerned, they are randomly generated. For the random generation of time window constraints, we, first, select the percentage of customers to receive time windows; then, the actual customers are randomly generated. The time windows have randomly generated centers and widths. The problem is consists of thirty customers shown in Table 2. The earliest and the latest service times are transformed into lengths from origin.

Figure 3 shows the shape of the solution of the problem where the percentage of customers to receive time windows was 50. In the case of thirty

Table 1 Comparison of solutions by various calculation methods.

<table>
<thead>
<tr>
<th></th>
<th>Best solution</th>
<th>Average</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>This algorithm</td>
<td>3754</td>
<td>4239</td>
<td>306</td>
</tr>
<tr>
<td>Sweep method +</td>
<td>3973</td>
<td>3978</td>
<td>15</td>
</tr>
<tr>
<td>this algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saving method</td>
<td>4297</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saving method +</td>
<td>3696</td>
<td>3783</td>
<td>81</td>
</tr>
<tr>
<td>this algorithm</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
customers, we could obtain a solution in a few ten seconds. The accuracy of the solution is now under investigation.

6. CONCLUSION

Above results revealed that the decentralized probabilistic algorithm using multi-agent system combined with the saving algorithm enables us to lead to fairly good solutions for the vehicle routing problems with a large-volume-customer.

This algorithm is also useful for the problems with time window constraints, and relatively small number of customers. But, it is necessary to investigate the accuracy of the solutions, and more complicate cases with vehicle capacities, various tightness of time window constraints. Further constraints are:

1) Demand may change during operation by generating new customers or disappearing existing customers.
2) Vehicles may have troubles, and an alternative scheduling should be programmed.

<table>
<thead>
<tr>
<th>Table 2 Problem set of 30 customers.</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
<tr>
<td>28</td>
</tr>
<tr>
<td>29</td>
</tr>
<tr>
<td>30</td>
</tr>
</tbody>
</table>

Nomenclature

$c_j$ A customer
$c_0$ Depot
$d_j$ Demand of a customer $c_j$
$v_i$ A vehicle
$V$ Set of vehicles
$C$ Set of customers
$u_i$ Capacity of vehicle $v_i$
$e_i$ Maximum travel cost for customer $c_i$
$l_{jk}$ Euclidean length between customer $c_j$ and customer $c_k$
$f(l_{jk})$ Cost proportional to the length $l_{jk}$
$g(d_j)$ Cost for loading and unloading the demand $d_j$
$x_{ij}^I$ 0-1 variable depending on a visit to customer $c_k$ immediately after customer $c_j$
$h'$ Sum of transportation cost and loading-unloading cost for the agent $I$
$a$ Constant
$t$ Time
$S^I(t)$ A set of customers assigned to $v_i$
$Q^I_s(t)$ Adding probability of a customer $c_k$ to $S^I(t)$, assigned to partial solutions of agents except $v_i$
$\eta_{ij}$ Inverse of length $l_{jk}$
$E(t)$ Evaluation function of solution
$\Delta E(t)$ Difference between values of $E(t)$ for new and the last solutions
$T_{max}$ Repeated time
$R(t)$ Acceptance probability of a new solution
$T$ Constant

151
References


