Heuristic Algorithm for Min-max Vehicle Routing Problems

Chunyu Ren
Heilongjiang University /School of Information science and technology, Harbin, China
Email: rency2004@163.com

Abstract—In order to satisfy with the individual and various demand of customer, the present study is focused on the Min-Max Vehicle Routing Problem (MMVRP). According to the characteristics of model, new tabu search algorithm is used to get the optimization solution. It applies newly improved insertion method to construct initial solution, to improve the feasibility of the solution; centers the longest route to design dual layered random operation to construct its neighborhood, to realize the strategy of optimizing between and within the route, to boost the efficiency and quality of the searching. Applies auto adaptive tabu length to control the searching capability dynamically; At last, it uses simulated experiments to prove the effectiveness and feasibility of this algorithm, and provides clues for massively solving practical problems.

Index Terms—MMVRP; new tabu search algorithm; improved insertion method; dual layered random operation; auto adaptive tabu length

I. INTRODUCTION

Vehicle Routing Problem is a typical NP problem. Its solution can be divided into three categories: the accurate algorithm, the classic heuristic algorithm and the modern intelligent algorithm. With respect to solving the practical large scaled problem, the heuristic algorithm can make it to get satisfactory solution in lesser time. Mester integrated the advantages of oriented local search and evolution strategies, designed an evolution strategy with positive guidance to solve VRP [1]. Derigs applied evolution strategies, designed an evolution strategy with integrated the advantages of oriented local search and tabu search algorithm to get the solution [2].

When it comes to solve massive complicated problems, the intelligent algorithm has wider application. Yuvraj applied large-scale ant colony algorithm to solve VRPB [3]. Baker conducted a research on VRP with indefinite number of vehicles and used improved genetic algorithm to solve it [4]. Gehring designed two-phase heuristic algorithm based on three strategies parallel to solve VRPTW [5]. Li H used insertion algorithm to construct initial solution, sequenced the customers through commuting operator, and then used simulated annealing algorithm to finish the control mentioned above [6]. Jose combined TS with Evolution Strategies to construct two-phase algorithm in solving VRPTW [8]. Ai designed a particle swarm optimization algorithm based on multiple social structures to solve pickup-delivery problem [9].

In practice, there exists a type of problems, whose aim is not to demand the shortest distance or the cheapest expenditure of the whole route, but to demand the shortest distance or the shortest time of the longest sub route throughout the whole route, for which is called Min-Max Vehicle Routing Problem, MMVRP.

Michael firstly solved the minimum boundary value of the objective function in MMVRP, and then used tabu search algorithm to get the solution [10]. Sema studied the school commuting bus MMVRP, and used the tabu search algorithm of the Cross and Or-opt Exchange Algorithm [11]. Arkin divided the n number of routes created by MMVRP into n number of sub regions of solving TSP problem and applied approximate algorithm to get the solution [12].

Considering the complexity of MMVRP, the essay proposed to apply new tabu search algorithm. Experiments proved that this algorithm can achieve not only better calculating results, but also better calculation efficiency and quicker convergence rate.

II. MATHEMATICAL MODEL

\[
Z = \text{Min} \left\{ \sum_{i \in S} \sum_{j \in S} \sum_{k \in V} X_{ijk} d_{ij} \right\}
\]

Constraints:

\[ \sum_{k \in V} Y_{ik} = 1, \quad i \in H \]  
(2)

\[ \sum_{i \in H} \sum_{j \in S} q_j X_{ijk} \leq W_k, \quad k \in V \]  
(3)

\[ \sum_{i \in S} X_{ijk} = Y_{ik}, \quad j \in S, \quad k \in V \]  
(4)

\[ \sum_{j \in S} X_{ijk} = Y_{ik}, \quad i \in S, \quad k \in V \]  
(5)

\[ \sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |V| - 1, \forall m \subseteq \{2, 3, \ldots, n\}, \quad k \in V \]
(6)

Corresponding Author: Chunyu Ren, email: rency2004@163.com; Tel: +88645186849571

© 2012 ACADEMY PUBLISHER
doi:10.4304/jcp.7.4.923-928
\[ \sum_{i \in V} \sum_{j \in H} X_{ij}d_{ij} \leq D_k, \quad j \in H \]  \hspace{1cm} (7)

Decision Variable:

- \( X_{ij} \) = 1, if travel vehicle \( k \) moves from node \( i \) to node \( j \), \( i \neq j \in S, k \in V \); or \( X_{ij} = 0 \).

- \( Y_k \) = 1, if travel vehicle \( k \) belongs to node \( i, i \in H, k \in V \), or \( Y_k = 0 \).

In the formula: \( G |g, p = 1,...,R| \) is a series of aggregations of distribution centre in the place \( R \) (this essay only has one); \( H |k, q = 1,...,N| \) is a series of clients’ aggregations in the place \( N \); \( S(G \cup H) \) is the combination of all distribution centers and clients. \( V |k, p = 1,...,K| \) is travel vehicle \( k \)'s aggregation; \( q_k \) is the demand amount of client \( i (i \in H) \); \( W_k \) is travel vehicle \( k \)'s loading capacity; \( d_{ij} \) is the linear distance from client \( i \) to client \( j \); \( D_k \) is the travel vehicle \( k \)'s maximum travel mileage.

In the formula (1), the objective function is not to demand the shortest distance of routes in the whole circuit, but to demand the shortest distance of the longest route in the whole circuit; constraint (2) ensures every client to be served by only one travel vehicle of one type; constraint (3) is for travel vehicle’s loading capacity, in order to ensure every vehicle in each route not to surpass its loading capacity; constraint (4) ensures every vehicle to reach the client and the distribution centre only once; constraint (5) ensures certain vehicle to be dispatched only once from certain distribution centre or certain receiving point; constraint (6) ensures the routes’ connectivity, that is to say, once the vehicle arrives at a client node, it must leave from the node; constraint (7) is for travel vehicle’s mileage, in order to ensure every vehicle in each route not to surpass its maximum travel mileage.

III. PARAMETER DESIGN FOR TABU SEARCH ALGORITHM

A. The formation of initial solution

Given \( h_k \) as the total number of client nodes served by vehicle \( k \), aggregation \( R_k = \{ y_{ik} | 0 \leq i \leq h_k \} \) to correspond the client nodes served by the number \( k \) vehicle, \( Y_{ik} \) signified that vehicle \( k \) served in node \( i \), \( Y_{0k} \) signified that the number \( k \) vehicle’s beginning point was distribution centre. The procedures as such: Step1: Order vehicles’ initial remaining load capacity: \( w_k^i = w_k^0, k = 0, h_k = 0, R_k = \Phi \).

Step2: The demand amount corresponding to the \( i \) client node in a route \( d_{ij} \), order \( k = 1 \).

Step3: if \( q_i \leq w_k^i \), then order \( w_k^i = Min \{ (w_k^i - q_i), w_k^i \} \), if not turn to Step6;

Step4: if \( w_k^i - q_i \leq w_k^i \), and \( D_{r+1} + D_i \leq D_k \); then \( R_k = R_k \cup \{ i \}, h_k = h_k + 1 \) if not turn to Step6;

Step5: if \( k > K \), then \( k = K \), otherwise, \( k = k \);

Step6: \( k = k + 1 \), turn to Step3;

Step7: \( i = i + 1 \), turn to Step2;

Step8: repeat Step2-7, \( K \) recorded the total used vehicles, \( R_k \) recorded a group of feasible routes.

B. Inner Neighborhood Operation

Specific procedures as such:

1. \( 1-move \)

\( 1-move \) is a heuristic algorithm the same as operators (1, 0) and (0, 1), which can effectively improve the quality of solutions and the feasibility of poor solutions. Specific operations are: delete a client point in a route, and then insert the client point into another route.

2. \( 2-opt \)

In the design of tabu search algorithm, \( 2-opt \) neighborhood operation was applied in the points in the same route, which was to exchange the service order between the two client points in the route in order to realize optimization in the route.

\( 2-opt \) is used to conduct neighborhood search, which was to randomly choose the positions of two client nodes, and then exchange the clients between the two positions. \( k(i) \) signified the neighbor point of the client point \( i \) in the route \( j \), and \( a(i, j) \) signified to change the direction of the route from \( i \) to \( j \). That was in the \( l \) route, the client points were: \( (0,1,2,...,n,0) \), in it, 0 signified distribution centre. The procedures of the \( 2-opt \) neighborhood operation were as such:

Step1: \( i_1 := 1, i_0 := 0 \);

Step2: if \( i > n - 2 \), end; otherwise, turn to Step3;

Step3: revise \( i_1 := k(i_1), j_1 := k(i_2), j := i + 2 \);

Step4: if \( j > n \), turn to Step8, if not, turn to Step5;

Step5: \( j_2 := s(j_1) \), change route \( l \) as such:

(1) \( a(i_2, j_1) \), (2) alternately used \( (i_1, j_1) \) and \( (i_2, j_2) \), substitute \( (i_1, j_1) \) and \( (i_2, j_2) \);

Step6: If the changed route \( l \) is feasible, and better than \( l \), revise \( l \), if not, turn to Step7;

Step7: \( j_1 := j_2, j := j + 1, \) return to Step4;

Step8: \( i_1 := i_2, \) \( i := i + 1, \) return to Step2.

C. Outer Neighborhood Operation

Specific procedures as such:

1. \( 1-exchange \)

\( 1-exchange \) method is to delete two clients in two routes, alternately insert them into their counterpart route, which can effectively boost the local search capability.
Its neighborhood structure is the same as \( 1 - move \), but its radius can be larger.

(2) \( 2 - opt' \)

\( 2 - opt' \) operates on the exchange of two edges in different routes, in order to realize optimization between routes. That is in the route \( l \), the client points are \( (0,1,2,...,n,0) \), in the route \( k \), the client points are \( (0,1,2,...,m,0) \), in it, 0 signifies distribution centre.

Step1: Randomly choose \( n \) number of client points in the route \( l \), for each client point \( i \), choose client point \( j \) nearby the route \( k \), if exist, exchange chains \( (i, i+1), (j, j+1) \);

Step2: Conduct \( 2 - opt \) neighborhood operation in the exchanged routes \( l' \) and \( k' \), to obtain feasible solution;

Step3: Calculate the exchanged objective function \( f' \), if \( f' > f \), turn to Step4; if not, turn to Step5;

Step4: If the current optimal solution does not exist in the tabu list, update tabu list, input the obtained optimal solution into the tabu list, simultaneously remove out the ban-lifted elements; otherwise, turn to Step5;

Step5: \( i = i + 1 \), turn to Step1;

Step6: repeat Step1-5, till the current optimal solution can not update.

D. Tabu List

Tabu list is a structure of tabu objective, which recording the process of reaching optimal solution as part in some iterative or solving.

The study establishes \((n+1) \times (n+1)\) rank matrix \( T \) to record the route reaching to tabu solution.

\[
T = \begin{bmatrix}
0 & a_1 & a_2 & \ldots & a_n \\
0 & a_1 & t_{11} & t_{12} & \ldots & t_{1n} \\
0 & a_1 & t_{21} & t_{22} & \ldots & t_{2n} \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & a_n & t_{n1} & t_{n2} & \ldots & t_{nn}
\end{bmatrix}
\]

Here, \( t_{ij} \) is expressed by number

\[
\begin{cases}
\text{forbiding exchange} & t_{ij} > 0 \\
\text{permitting exchange} & t_{ij} = 0
\end{cases}
\]

E. Adaptive Tabu Length

Tabu length is a crucial parameter in the algorithm, and the length of its tenure determines the solution’s option. Lesser the tenure, larger the possibility of obtaining better solution, but at the same time circuitous search is added, and if its tenure over prolongs, more calculating time will be added. In order to ensure effectiveness of the tabu list, during the whole process of searching, make \( L \) as its variable region \([a\sqrt{N}, b\sqrt{N}]\), in it \( 0 < a < b \). So the tabu length \( L \)’s variable scope is the formula as the following:

\[
L = \lambda L_{\text{min}} + (1 - \lambda)L_{\text{max}}
\]

In the formula, \( L_{\text{min}} \) and \( L_{\text{max}} \) are the upper and lower bound of tabu length \( L \)’s dynamic change respectively, \( N \) refers to the number of clients, the weighing coefficient is \( 0 \leq \lambda \leq 1 \).

F. Mountain climbing operation

Add mountain climbing algorithm into genetic algorithm. Improve the solution of each generation structure so as to reduce the expense of solving route and quicken algorithm constringency speed. Mountain climbing algorithm adopts 2-opt method to take point exchanging operation.

Supposed routing line before exchanging is \( s = \{...,x_i,x_{i+1},...,x_j,x_{j+1},...,\} \), it can get the routing line \( s' = \{...,x_i,x_{i+1},...,x_j,x_{j+1},...,\} \) after exchanging location of two points. If the distance is equal to \( d(x_{i+1},x_i) + d(x_j,x_{j+1}) < d(x_{i+1},x_i) + d(x_j,x_{j+1}) \), exchanging is successful and keeps exchanging result. Otherwise, exchanging attempt is failure, cancel exchanging and furnish former routing line.

To the optimized individual of each generation group through genetic operation, it can realize mountain climbing operation through searching in neighbors. The study adopts gene exchanging operator to realize climbing operation. The concrete steps are as followsings.

Step1: Initial recycling time variable \( t = 1 \), when the most optimal solution at present \( s' = s \) and its length \( l(s') \).

Step2: Randomly selecting two top points \( x_i, x_j \) in the most optimal route and \( i < j \). \( x_i \) is not close to \( x_j \).

Step3: Calculate saving distance,

\[
\Delta x = [d(x_{i+1},x_i) + d(x_j,x_{j+1})] < [d(x_{i+1},x_i) + d(x_j,x_{j+1})]
\]

If \( \Delta x > 0 \) , it isn’t exchanged. If \( t = t + 1 \), it shifts into step4.

Otherwise, execute exchanging. And the corresponding solution is \( s' \). And the optimal solution is \( s' = s' \). If \( t = 1 \), it shifts into step 2.

Step4: If \( l(s') \) isn’t reduced in the last x a circulation, this algorithm is over. Otherwise, it shifts into step 2.

Step5: Repeat step 1 to step 4 till reaching certain exchanging times.

G. The procedures of the algorithm

The procedures of hybrid heuristic algorithm based on tabu search algorithm are as such:

Step1: Give the parameters of the algorithm: initial tabu list \( T[T[i,j] = 0] \), the current iteration number \( \text{iter} = 0 \), the current continuous iteration
number of not finding the best solution unchange_iter = 0, the initial feasible solution $R_{row}$, the current best solution $R_{best}$, and the current best function value $f^{*} = f(R_{best})$;

Step2: if max_iter = iter or unchange_iter = max_unchange_iter, stop calculating, turn to Step9, otherwise, and continue;

Step3: In the neighborhood of initial feasible solution $R_{row}$, alternately conduct 1-move and 2-opt operations, choose a non-tabooed or amnestied and best value-assessed solution $R_{next}$ in this neighborhood, order $R_{row} = R_{next}$, and update tabu list $T$;

Step4: In the neighborhood of initial feasible solution $R_{row}$, randomly conduct 1-exchange and 2-opt operations choose a non-tabooed or amnestied solution with better assessment value than $R_{next}$, order $R_{row} = R_{next}$, and update tabu list $T$;

Step5: Repeat Step3-4, till all the changes in the $R_{row}$ neighborhood is tabooed and none is amnestied;

Step6: In the $R_{row}$ neighborhood, search non-tabooed optimal candidate solution or the best solution better than the current one, if the objective function $f(R_{row})$ is better than $f(R_{best})$, order, $R_{best} = R_{row}$, unchange_iter = 0, turn to Step8;

Step7: if $f(R_{row})$ is not better than $f(R_{best})$, unchange_iter = unchange_iter + 1, turn to Step8;

Step8: update tabu list $T$, iter = iter + 1 turn to Step2;

Step9: Output the optimal solution $f^{*}$.

IV. EXPERIMENTAL CALCULATION AND RESULT ANALYSIS

Example one: The data originates from Document [14]. There are one depot and 20 client nodes, the coordinates and demand amount of each node is created randomly, as indicated in table 1 (the depot’s number is 0); give six vehicles of the same type, and the vehicle’s load capacity is 8.

A. Solution of New Tabu Search Algorithm

Tabu search algorithm adopts the following parameters as part. The maximum iterative times are max_iter = 500, tabu length is $\alpha = 2$, $\beta = 3$, $\lambda = 0.6$, and candidate solution amount is 50. Randomly solve ten times and calculation results can be seen as table 2.

<table>
<thead>
<tr>
<th>Item</th>
<th>coordinate</th>
<th>Distribution amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(52, 4)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>(13, 49)</td>
<td>1.64</td>
</tr>
<tr>
<td>2</td>
<td>(0, 61)</td>
<td>1.31</td>
</tr>
<tr>
<td>3</td>
<td>(51, 15)</td>
<td>0.43</td>
</tr>
<tr>
<td>4</td>
<td>(25, 71)</td>
<td>3.38</td>
</tr>
<tr>
<td>5</td>
<td>(38, 62)</td>
<td>1.13</td>
</tr>
<tr>
<td>6</td>
<td>(35, 45)</td>
<td>3.77</td>
</tr>
<tr>
<td>7</td>
<td>(100, 4)</td>
<td>3.84</td>
</tr>
<tr>
<td>8</td>
<td>(0, 10, 52)</td>
<td>0.39</td>
</tr>
<tr>
<td>9</td>
<td>(26, 79)</td>
<td>0.24</td>
</tr>
<tr>
<td>10</td>
<td>(87, 7)</td>
<td>1.03</td>
</tr>
<tr>
<td>11</td>
<td>(24, 89)</td>
<td>2.35</td>
</tr>
<tr>
<td>12</td>
<td>(19, 25)</td>
<td>2.60</td>
</tr>
<tr>
<td>13</td>
<td>(20, 99)</td>
<td>1.00</td>
</tr>
<tr>
<td>14</td>
<td>(73, 91)</td>
<td>0.65</td>
</tr>
<tr>
<td>15</td>
<td>(100, 25)</td>
<td>0.85</td>
</tr>
<tr>
<td>16</td>
<td>(7, 73)</td>
<td>2.56</td>
</tr>
<tr>
<td>17</td>
<td>(69, 86)</td>
<td>1.27</td>
</tr>
<tr>
<td>18</td>
<td>(24, 3)</td>
<td>2.69</td>
</tr>
<tr>
<td>19</td>
<td>(66, 14)</td>
<td>3.26</td>
</tr>
<tr>
<td>20</td>
<td>(9, 30)</td>
<td>2.97</td>
</tr>
</tbody>
</table>

It can be known that new tabu search algorithm in the study all get the much higher solution during the course of ten times from table 2. The average value of total distance is 1092.109(km) and the average using vehicles are six. The calculation result of algorithm is relatively steady. The total distance of least solution is only better 1.431 percent than the best.

<table>
<thead>
<tr>
<th>Calculation order</th>
<th>New Tabu Search Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total distance</td>
</tr>
<tr>
<td>1</td>
<td>1100.073</td>
</tr>
<tr>
<td>2</td>
<td>1095.813</td>
</tr>
<tr>
<td>3</td>
<td>1095.813</td>
</tr>
<tr>
<td>4</td>
<td>1095.136</td>
</tr>
<tr>
<td>5</td>
<td>1097.753</td>
</tr>
<tr>
<td>6</td>
<td>1100.073</td>
</tr>
<tr>
<td>7</td>
<td>1082.918</td>
</tr>
<tr>
<td>8</td>
<td>1087.674</td>
</tr>
<tr>
<td>9</td>
<td>1095.813</td>
</tr>
<tr>
<td>10</td>
<td>1082.918</td>
</tr>
<tr>
<td>Average value</td>
<td>1092.109</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>7.214</td>
</tr>
</tbody>
</table>
B. Solutions by genetic algorithm

Reference [14] is adopted genetic algorithm to get the solution.

The main parameters: population size of 50, the maximum number of iterations is 400; crossover 0.80, mutation operator is 0.2. The concrete route can be seen in table 4 and figure 2.

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Running Path</th>
<th>Mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-1-8-16-19-0</td>
<td>185.945</td>
</tr>
<tr>
<td>2</td>
<td>0-9-13-11-4-0</td>
<td>201.293</td>
</tr>
<tr>
<td>3</td>
<td>0-18-3-7-10-0</td>
<td>156.254</td>
</tr>
<tr>
<td>4</td>
<td>0-6-5-14-17-0</td>
<td>197.247</td>
</tr>
<tr>
<td>5</td>
<td>0-15-0</td>
<td>205.767</td>
</tr>
<tr>
<td>6</td>
<td>0-12-20-2-0</td>
<td>159.731</td>
</tr>
</tbody>
</table>

The Total Mileage: 1106.237 km
Average Mileage: 184.373 km
The longest line: 205.767 km

C. Solutions by Tabu Search Algorithm

Reference [14] is adopted tabu search algorithm to get the solution.

The main parameters: the size of the neighborhood list of 20, the candidate set of 10, taboo list is 10, and end time is 5 seconds. The concrete route can be seen in table 5 and figure 3.

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Running Path</th>
<th>Mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-16-2-8-20-0</td>
<td>181.416</td>
</tr>
<tr>
<td>2</td>
<td>0-9-11-13-4-0</td>
<td>201.293</td>
</tr>
<tr>
<td>3</td>
<td>0-18-10-7-0</td>
<td>132.486</td>
</tr>
<tr>
<td>4</td>
<td>0-6-5-17-14-0</td>
<td>196.754</td>
</tr>
<tr>
<td>5</td>
<td>0-15-0</td>
<td>205.767</td>
</tr>
<tr>
<td>6</td>
<td>0-12-1-3-19-0</td>
<td>145.202</td>
</tr>
</tbody>
</table>

The Total Mileage: 1082.918 km
Average Mileage: 180.486 km
The longest line: 205.767 km
D. Analysis on Three Algorithms

Compared the optimal scheme of reference [14], the proposed hybrid genetic algorithm has a strong search capability, high computational efficiency and high quality on algorithm solving.

<table>
<thead>
<tr>
<th>Line No.</th>
<th>Running Path</th>
<th>Mileage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-18-10-7-0</td>
<td>132.486</td>
</tr>
<tr>
<td>2</td>
<td>0-3-9-16-2-8-20-0</td>
<td>199.298</td>
</tr>
<tr>
<td>3</td>
<td>0-15-0</td>
<td>205.767</td>
</tr>
<tr>
<td>4</td>
<td>0-4-13-11-5-0</td>
<td>201.529</td>
</tr>
<tr>
<td>5</td>
<td>0-12-1-19-0</td>
<td>142.506</td>
</tr>
<tr>
<td>6</td>
<td>0-6-17-14-0</td>
<td>193.550</td>
</tr>
</tbody>
</table>

The Total Mileage 1095.136 km
Average Mileage 182.523 km
The longest line 205.767 km

V. CONCLUSIONS

In general, the proposed new tabu search algorithm has strong searching ability, rapid convergence rate, strong ability to overcome the fall into local optimum and high solving high quality. Therefore, it is more practical significance and value so as to reduce operating cost and improve economic benefit.

ACKNOWLEDGMENT

This paper is supported by project of National Social Science Foundation (No. 10CGL076), Natural Science Foundation of Heilongjiang Province (NO. G201020) and Science & Technology Foundation of Heilongjiang Provincial Education Department (No. 11551332)

REFERENCES


Chunyu Ren is an associate professor in the department of school of information science and technology, Heilongjiang University, Harbin, China. She holds a master degree in Computer Science and Technology from the Harbin Institute of Technology, Harbin, China. Her previous research areas include e-commerce, logistics system simulation, applications of artificial optimization algorithm and business intelligence. E-mail: rency2004@163.com

© 2012 ACADEMY PUBLISHER