The Split Delivery Min-Max Multi-Depot Vehicle Routing Problem with Minimum Service Time Requirement

Xingyin Wang, Bruce Golden, and Edward Wasil

POMS Washington DC

May 8, 2015
Overview

Introduction

Algorithm
  Overview of the algorithm
  Cyclic transfer operator

Computational results
  New test instances
  Estimated solutions
  Performance

Conclusions
Introduction: Min-max objective

- In the standard Multi-Depot VRP, the objective is to minimize the total distance traveled by all vehicles.

- In the Min-max Multi-Depot VRP, the objective is to minimize the maximum distance traveled by a vehicle (Carlsson et al. 2009).
Introduction: Min-max objective

Why is the min-max objective important?

- Disaster relief efforts
  Serve all victims as soon as possible
- Computer networks
  Minimize the maximum latency between the server and a customer
- Workload balance
  Balance workload among drivers or across the time horizon
Introduction: Split service

- Yakici and Karasakal (2013) studied a min-max service VRP with split delivery and heterogeneous demand

- Duration of a route = travel time + service time

- Service times can significantly change the optimal routing plan of the min-max VRP (Bertazzi et al. 2014)
Introduction: Minimum delivery

- Split delivery may inconvenience the customers
- Gulczyński et al. (2010) introduced a split delivery VRP with minimum delivery amounts
Introduction

We want to develop an algorithm for a problem with

- Min-max objective
- Multiple depots
- Service times
- Split deliveries
- Minimum service time requirement
Algorithm: Overview

We develop an heuristic algorithm (MDS) that has three stages

1. Initialization (Wang et al. 2015)
   - load balancing (Carlsson et al. 2009)
   - local search without split service
   - perturbation without split service
   - cyclic transfer
   - one-point move w.r.t. the min-sum objective
   - two-point move w.r.t. the min-sum objective
   - route destruction and re-construction
Algorithm: Overview

2. Improvement
   • local search with split delivery
   • perturbation with split delivery

3. Post-process for minimum service time requirement
Algorithm: Cyclic transfer operator

(a) Before cyclic transfer

(b) After cyclic transfer

Cyclic transfer
Algorithm: Cyclic transfer operator

- Thompson and Psaraftis (1993)
- Notation
  - $R_j$: route that serves customer $j$
  - $R_j - j$: route formed by removing customer $j$ from $R_j$
  - $R_j - j + i$: route formed by adding customer $i$ to $R_j - j$
  - $D(R_j - j + i)$: duration of the route $R_j - j + i$
  - $Z$: longest duration of the solution before cyclic transfer
Algorithm: Cyclic transfer operator

1. Auxiliary graph generation
   - Node $i$: customer $i$
   - Arc $(i, j)$: $R_i \neq R_j$ and $D(R_j - j + i) < Z$
     - Cost of arc $(i, j) = 0$ if $D(R_j) < Z$
     - Cost of arc $(i, j) = -1$ if $D(R_j) = Z$

2. Strongly connected components and negative cycles

3. First profitable cycle
Algorithm: Cyclic transfer operator

(c) Before cyclic transfer

(d) Auxiliary graph

Cyclic transfer
Algorithm: Cyclic transfer operator

(e) Before cyclic transfer

(f) After cyclic transfer

Cyclic transfer
Computational Results: New test instances

- Chen et al. (2007)
- \( n = A \times B \) customers located on concentric circles centered at the depot
  - \( A \): number of customers per circle
  - \( B \): number of circles
  - Difference in radii between adjacent circles is 100
- Customer service time is 100
- Number of vehicles is \( m = \frac{3A}{2} \)
Computational Results: New test instances

- 21 instances with different values for $A$ and $B$

(g) $A = 16, B = 2, m = 24$

(h) $A = 16, B = 4, m = 24$

SD6 and SD10
Computational Results: Estimated solutions

Partition on SD10 (\(A = 16\), \(B = 4\), and \(m = 24\))

Subproblem of SD10

- Partition the region into \(\frac{A}{2}\) equal sectors
- Solve the subproblem on each partition with \(2B\) customers and 3 vehicles
Computational Results: Estimated solutions

- First route reaches the farthest customer on the first ray
- Second route reaches the $k^{th}$ customer on the first ray, then visits the $k^{th}$ customer on the second ray before returning to the depot
- Third route reaches the farthest customer on the second ray
Computational Results: Estimated solutions

Proposed estimated solution to the subproblem

- $R_1$: 0 – 49 – 33 – 0
  Duration = 1000
- $R_2$: 0 – 1 – 17 – 32 – 16 – 0
  Duration = 879.75
- $R_3$: 0 – 64 – 48 – 0
  Duration = 1000
- $k = 2$
Computational Results: Estimated solutions

Proposed estimated solution to the subproblem

<table>
<thead>
<tr>
<th>Duration</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 3$</th>
<th>$k = 4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1$</td>
<td>1100</td>
<td>1000</td>
<td>1040</td>
<td>1120</td>
</tr>
<tr>
<td>$R_2$</td>
<td>440</td>
<td>880</td>
<td>1040</td>
<td>1120</td>
</tr>
<tr>
<td>$R_3$</td>
<td>1100</td>
<td>1000</td>
<td>1040</td>
<td>1120</td>
</tr>
</tbody>
</table>

Route duration for different $k$ values
## Computational Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n$</th>
<th>Exact</th>
<th>Estimated</th>
<th>MDS</th>
<th>Gap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD1</td>
<td>8</td>
<td>513.81</td>
<td>513.81</td>
<td>513.81</td>
<td>0.00</td>
</tr>
<tr>
<td>SD2</td>
<td>16</td>
<td>1027.61</td>
<td>1027.61</td>
<td>1027.61</td>
<td>0.00</td>
</tr>
<tr>
<td>SD3</td>
<td>16</td>
<td>500.00</td>
<td>500.00</td>
<td>500.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SD4</td>
<td>24</td>
<td>500.00</td>
<td>500.00</td>
<td>500.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SD5</td>
<td>32</td>
<td>1000.00</td>
<td>1000.00</td>
<td>1000.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SD6</td>
<td>32</td>
<td>−</td>
<td>500.00</td>
<td>500.00</td>
<td>0.00</td>
</tr>
<tr>
<td>SD7</td>
<td>40</td>
<td>−</td>
<td>2569.04</td>
<td>2600.00</td>
<td>1.21</td>
</tr>
<tr>
<td>SD8</td>
<td>48</td>
<td>−</td>
<td>3082.84</td>
<td>3100.00</td>
<td>0.56</td>
</tr>
<tr>
<td>SD9</td>
<td>48</td>
<td>−</td>
<td>1000.00</td>
<td>1026.28</td>
<td>2.63</td>
</tr>
<tr>
<td>SD10</td>
<td>64</td>
<td>−</td>
<td>1000.00</td>
<td>1017.20</td>
<td>1.72</td>
</tr>
<tr>
<td>SD11</td>
<td>80</td>
<td>−</td>
<td>5100.00</td>
<td>5100.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Exact, estimated, and MDS solutions
## Computational Results

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n$</th>
<th>Exact</th>
<th>Estimated</th>
<th>MDS</th>
<th>Gap(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD12</td>
<td>80</td>
<td>–</td>
<td>2500.00</td>
<td>2583.79</td>
<td>3.35</td>
</tr>
<tr>
<td>SD13</td>
<td>96</td>
<td>–</td>
<td>3000.00</td>
<td>3012.49</td>
<td>0.42</td>
</tr>
<tr>
<td>SD14</td>
<td>120</td>
<td>–</td>
<td>2500.00</td>
<td>2553.28</td>
<td>2.13</td>
</tr>
<tr>
<td>SD15</td>
<td>144</td>
<td>–</td>
<td>2987.45</td>
<td>3067.69</td>
<td>2.69</td>
</tr>
<tr>
<td>SD16</td>
<td>144</td>
<td>–</td>
<td>500.00</td>
<td>503.21</td>
<td>0.64</td>
</tr>
<tr>
<td>SD17</td>
<td>160</td>
<td>–</td>
<td>5000.00</td>
<td>5023.53</td>
<td>0.47</td>
</tr>
<tr>
<td>SD18</td>
<td>160</td>
<td>–</td>
<td>2478.04</td>
<td>2600.00</td>
<td>4.92</td>
</tr>
<tr>
<td>SD19</td>
<td>192</td>
<td>–</td>
<td>2957.71</td>
<td>3054.28</td>
<td>3.27</td>
</tr>
<tr>
<td>SD20</td>
<td>240</td>
<td>–</td>
<td>4923.13</td>
<td>5119.63</td>
<td>3.99</td>
</tr>
<tr>
<td>SD21</td>
<td>288</td>
<td>–</td>
<td>1000.00</td>
<td>1034.09</td>
<td>3.49</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>1.50</strong></td>
</tr>
</tbody>
</table>

Exact, estimated, and MDS solutions
Conclusions

• Developed a heuristic (MDS) that solved the Min-max Split Delivery Multi-Depot VRP with Minimum Service Time Requirement

• Constructed a family of instances with estimated solutions

• Tested MDS on these instances and showed that the average gap from the estimated solutions was 1.5%
Q & A

wangxy@umd.edu

bgolden@rhsmith.umd.edu

ewasil@american.edu