A Tabu Search Algorithm for the Vehicle Routing Problem with Deliveries and Pickups

Niaz Wassan and Gábor Nagy

Centre for Heuristic Optimisation
Kent Business School
University of Kent

Vehicle Routing with Pickups and Deliveries

Problem Definition
Two types of customers
- Linehauls: expect delivery of goods from a depot
- Backhauls: wish to send goods to the depot

Objective: find minimum-cost set of vehicle routes
Constraints: Satisfy all customer demand
- Respect maximum capacity of vehicles

We assume all goods go to, or come from, the depot — that is, no goods are transported direct from one customer to another.

Applications

Delivery of drink bottles to shop, pickup of empty bottles
Delivery and pickup of mail to/from customers or post offices
Delivery of new household appliances and removal of old ones
Airport minibus: transporting travellers between home/hotel and airport
Problem Versions

1. backhaul (VRPB)
   all linehauls must be served before backhauls

2. mixed (VRPMDP)
   linehauls and backhauls can occur in any order

3. simultaneous (VRPSDP)
   customers can have both demand and supply

4. other
   not all pickup and delivery problems are VRPDP

Literature Review

Although vehicle routing has a wide subject literature (see e.g. Toth and Vigo, book, 2002), vehicle routing with pickups and deliveries attracted relatively little attention, but developed steadily of late...

<table>
<thead>
<tr>
<th>classical backhaul</th>
<th>mixed and simultaneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goetschalckx et al. (EJOR, 1989)</td>
<td>Dethloff (Spektrum, 2001)</td>
</tr>
<tr>
<td>Toth and Vigo (TranspSci, 1997)</td>
<td>Wade and Salhi (Omega, 2002)</td>
</tr>
<tr>
<td>Toth and Vigo (EJOR, 1999)</td>
<td>Nagy and Salhi (EJOR, 2005)</td>
</tr>
<tr>
<td>Osman and Wassan (JSched, 2002)</td>
<td>Crispim and Brandão (JORS, 2005)</td>
</tr>
<tr>
<td>Wade and Salhi (Omega, 2002)</td>
<td>Tang and Galvão (CompOR, 2006)</td>
</tr>
<tr>
<td>Wassan (JORS, 2007)</td>
<td>Chen and Wu (JORS, 2006)</td>
</tr>
</tbody>
</table>

Initial Solution Phase

Sweep Method proposed by Gillett and Miller (1974) to generate initial solution
   - this approach orders the customers according to their polar coordinates and then partitions the ordering into feasible routes

Modified Sweep
   - customers near the depot are excluded from sweep
     - they form their own single-customer routes
     - this gives flexibility to the algorithm later on

Improvement Phase — various improvement routines
   - Shift: moves a customer from one route to another
   - Swap: swaps two customers that are on different routes
   - Local-shift: moves a customer to a different place on its route
   - Reverse: reverses route direction (can be helpful!)
Tabu Search

Proposed by Fred Glover

An extension to neighbourhood search, allowing non-improving moves. To avoid cycling, a “tabu list” is used to prevent returning to solutions already visited. The tabu list is updated from time to time with respect to “Tabu Tenure” ($tt$ value).

- Fixed $tt$
- Robust $tt$
- Reactive $tt$

Tabu search used by:
- Crispim and Brandão (JORS, 2005) — simultaneous/mixed
- Tang and Galvão (CompOR, 2006) — simultaneous
- Chen and Wu (JORS, 2006) — simultaneous

Reactive Tabu Search

Proposed by Battiti and Tecchioli (1994)

A refinement of tabu search where the length of time ($tt$ value) record remains in the tabu list is controlled dynamically.

Reactive tabu search used by:
- Osman and Wassan (JSched, 2002) — backhaul
- Wassan (JORS, to appear) — backhaul
- Wassan, Nagy and Wassan (in progress) — simultaneous/mixed

There are other ways of enhancing the tabu search framework:
Using both long- and short-term memory (Crispim and Brandão, JORS, 2005)
Combining it with adaptive memory programming (Wassan, JORS, to appear)

RTS Algorithm for VRPSDP

[Step 1] Initialisation
- Start with the constructed initial solution by the Modified Sweep method as $S_{best}$.
- Set stopping rule
- Initialise data structures for the tabu search components
- Initialise RTS parameters

[Step 2] Start search
- Perform shift and swap neighbourhood moves and complete a cycle of search
- Scan for the best neighbouring solution, $S'$
- Apply local shift and reverse post-optimizer procedures to get any further improvement on the current set of routes
- Update route configurations by setting $S = S'$ as a current solution

[Step 3] RTS mechanism
- Check if the current solution is a repetition
- If repetition was found, then increase $tt$ by 10%
- Else if repetition was not found, then decrease $tt$ by 10%

[Step 4] Update the search parameters and the best solution
- $iter = iter + 1$
- If $C(S) < C(S_{best})$ then set $S_{best} = S$
- Check for stopping rule; STOP
- Else go to [Step 2]
1. Computational Results – Backhaul Problems

Class 1 dataset — Goetschalckx & Jacobs-Blecha (EJOR, 1989) — 62 instances
25 to 150 customers; proportion of backhauls between 20% and 50%

Class 2 dataset — Toth and Vigo (TranspSci, 1997) — 33 instances
21 to 100 customers; proportion of backhauls is either 1/2, 1/3 or 1/5

reactive tabu adaptive memory programming (Wassan, JORS, to appear)
21 new best results for dataset 1 (real values)
improvement over Toth & Vigo (1996) \(1.13\)% — optimality gap \(0.02\)%
15 new best results for dataset 1 (integer values)
improvement over Toth & Vigo (1999) \(1.00\)%
9 new best results for dataset 2
improvement over Toth & Vigo (1999) \(0.77\)% — optimality gap \(0.76\)%

Computational Results – Mixed Problems

Class 1 dataset — Goetschalckx \textit{et al} (EJOR, 1989); Halse (PhD, 1992)
25 to 150 customers; proportion of backhauls between 20% and 50%

Class 2 dataset — Toth and Vigo (TranspSci, 1997); Wade (PhD, 2002)
21 to 100 customers; proportion of backhauls is either 1/2, 1/3 or 1/5

Class 3 dataset — Salhi and Nagy (JORS, 1999) — 28 instances
50 to 199 customers; proportion of backhauls is either a 1/10, 1/4 or 1/2
time constraints apply to half the instances

reactive tabu search (Wassan and Nagy, in review)
31 new best for dataset 1 — improved Halse by \(0.98\)%
15 new best for dataset 2 — improved Salhi \textit{et al} by \(0.55\)%
30 new best for dataset 3 — improved Crispim and Brandão by \(4.37\)%

Suggestions for Future Research

Technical enhancements
Combining reactive tabu search with adaptive memory programming
Combining reactive tabu with variable neighbourhood search
Allowing search to traverse infeasible solutions

Model extensions
Multiple depots
Heterogeneous fleet
Allowing mixture of linehaul and backhaul goods only subject to extra free space
Airport minibus: a stochastic-deterministic problem
### Table 6.1: Comparison between different algorithms for the CMT problems (without maximum distance constraints).

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Nagy &amp; Salhi</th>
<th>Dethloff</th>
<th>Crupianni &amp; Brand</th>
<th>Chen &amp; Wu</th>
<th>Tang &amp; Galvão</th>
<th>RTS-VRPSPD</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS-VRPSPD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sol</td>
<td>T</td>
<td>Sol</td>
<td>V</td>
<td>Sol</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td></td>
</tr>
<tr>
<td>CMT10 50</td>
<td>525</td>
<td>8.15</td>
<td>509</td>
<td>3.83</td>
<td>488.30</td>
<td>10.40</td>
<td>599.70</td>
</tr>
<tr>
<td>CMT10 55</td>
<td>525</td>
<td>8.15</td>
<td>509</td>
<td>3.83</td>
<td>488.30</td>
<td>10.40</td>
<td>599.70</td>
</tr>
<tr>
<td>CMT15 50</td>
<td>1020</td>
<td>15.50</td>
<td>1011</td>
<td>9.60</td>
<td>1004.40</td>
<td>19.20</td>
<td>1072.00</td>
</tr>
<tr>
<td>CMT15 55</td>
<td>1020</td>
<td>15.50</td>
<td>1011</td>
<td>9.60</td>
<td>1004.40</td>
<td>19.20</td>
<td>1072.00</td>
</tr>
<tr>
<td>CMT20 50</td>
<td>1980</td>
<td>30.10</td>
<td>1974</td>
<td>23.00</td>
<td>1910.80</td>
<td>35.70</td>
<td>2142.30</td>
</tr>
<tr>
<td>CMT20 55</td>
<td>1980</td>
<td>30.10</td>
<td>1974</td>
<td>23.00</td>
<td>1910.80</td>
<td>35.70</td>
<td>2142.30</td>
</tr>
<tr>
<td>CMT25 50</td>
<td>3960</td>
<td>50.30</td>
<td>3957</td>
<td>38.20</td>
<td>3735.60</td>
<td>58.30</td>
<td>4155.60</td>
</tr>
<tr>
<td>CMT25 55</td>
<td>3960</td>
<td>50.30</td>
<td>3957</td>
<td>38.20</td>
<td>3735.60</td>
<td>58.30</td>
<td>4155.60</td>
</tr>
</tbody>
</table>

**Average gap:** 15.1% 15.2% 15.3%

Sol: Solution value; V: number of vehicles used in the solution; LB: lower bound; T: CPU time in seconds; TB: CPU time in best; TT: Total CPU time

Best solutions are highlighted in bold.

Average gap is with respect to the lower bound (found by Tang & Galvão).

---

### Table 6.2: Comparison between different algorithms for the CMT problems (with maximum distance constraints).

<table>
<thead>
<tr>
<th>Problem Size</th>
<th>Nagy &amp; Salhi</th>
<th>Dethloff</th>
<th>Crupianni &amp; Brand</th>
<th>Chen &amp; Wu</th>
<th>Tang &amp; Galvão</th>
<th>RTS-VRPSPD</th>
<th>LB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTS-VRPSPD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sol</td>
<td>T</td>
<td>Sol</td>
<td>V</td>
<td>Sol</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td>(in sec)</td>
<td></td>
</tr>
<tr>
<td>CMT10 50</td>
<td>525</td>
<td>3.83</td>
<td>509</td>
<td>3.83</td>
<td>488.30</td>
<td>10.40</td>
<td>599.70</td>
</tr>
<tr>
<td>CMT10 55</td>
<td>525</td>
<td>3.83</td>
<td>509</td>
<td>3.83</td>
<td>488.30</td>
<td>10.40</td>
<td>599.70</td>
</tr>
<tr>
<td>CMT15 50</td>
<td>1020</td>
<td>9.60</td>
<td>1011</td>
<td>9.60</td>
<td>1004.40</td>
<td>19.20</td>
<td>1072.00</td>
</tr>
<tr>
<td>CMT15 55</td>
<td>1020</td>
<td>9.60</td>
<td>1011</td>
<td>9.60</td>
<td>1004.40</td>
<td>19.20</td>
<td>1072.00</td>
</tr>
<tr>
<td>CMT20 50</td>
<td>1980</td>
<td>18.20</td>
<td>1974</td>
<td>15.90</td>
<td>1910.80</td>
<td>27.10</td>
<td>2142.30</td>
</tr>
<tr>
<td>CMT20 55</td>
<td>1980</td>
<td>18.20</td>
<td>1974</td>
<td>15.90</td>
<td>1910.80</td>
<td>27.10</td>
<td>2142.30</td>
</tr>
<tr>
<td>CMT25 50</td>
<td>3960</td>
<td>26.30</td>
<td>3957</td>
<td>23.60</td>
<td>3735.60</td>
<td>39.30</td>
<td>4155.60</td>
</tr>
<tr>
<td>CMT25 55</td>
<td>3960</td>
<td>26.30</td>
<td>3957</td>
<td>23.60</td>
<td>3735.60</td>
<td>39.30</td>
<td>4155.60</td>
</tr>
</tbody>
</table>

**Previous best result:** from Chen & Wu (C), Tang & Galvão (T) or Dethloff (D).

**V:** number of vehicles used in the solution.

**TB:** CPU time (in seconds) to best; **TT:** Total CPU time

Single-run solutions that are better than previously known results are highlighted in bold. (This is not done for the instances with maximum distance constraints, as these are always better than the results of Dethloff.)

---

**Previous best result:** from Chen & Wu (C), Tang & Galvão (T) or Dethloff (D).

**V:** number of vehicles used in the solution.

**TB:** CPU time (in seconds) to best; **TT:** Total CPU time

Single-run solutions that are better than previously known results are highlighted in bold. (This is not done for the instances with maximum distance constraints, as these are always better than the results of Dethloff.)