MIRHA

Multi-start biased randomization of heuristics with adaptive local search for solving non-smooth routing problems

Angel A. Juan (UOC)
Javier Faulin (UPN)
Albert Ferrer (UPC)
Helena R. Lourenço (UPF)
Barry Barrios (MIT)

Outline of the Presentation

- Motivation and introduction
- VRP with non-smooth/non-convex objective functions
- MIRHA approach
- Computational results
- Conclusions and directions of future work.
Motivation & Introduction

► Non-convex optimization problems (NCOPs) the objective function or even the feasible region are not convex.
► NCOPs result in a far more complex solution space than in the case of COPs.
► Many real applications that need an efficient answer…
► But not as many publications as for the COPs.

Motivation & Introduction

► We present a simple -almost parameter free- and efficient methodology, which can provide pseudo-optimal solutions to difficult problems in reasonable computing times.
► The method will be evaluated by its:
  - Accuracy (quality of results)
  - Simplicity of design and implementation
  - Efficiency
  - Robustness and Flexibility
Example of a non-convex function

VRP with non-smooth/non-convex objective functions

- Vehicle Routing Problems are well-known problems with many real applications.
- But more real problems tend to be more complex than the classical VRP.
- There is the need of an efficient and robust approach to deal with these real and complex applications of the VRP.
VRP with non-smooth/non-convex objective functions

► Examples:
- Minimization of fuel consumptions in surface transportation.
  * costs related with roads slopes or types of asphalts, weather or temperature
- Minimization of CO2 emissions related to road transportation.
- Other costs, as penalty function for not complete cargo, time windows penalties, or drivers incentives. etc.
- Penalties for soft constraints.

VRP with non-smooth/non-convex objective functions

► Penalties for soft constraints
- We considered soft constraints, which allow conditions to be violated, by incurring some penalty costs that must be added to the objective function.
- Penalties for capacities and individual route costs associated with the cargo.
  * For each route, if the route cost is less than $C_{\text{max}}$, this is the route cost, otherwise a penalty function is applied.
  * non-linear and non-smooth function
VRP with non-smooth/non-convex objective functions

► Let $\rho$ be a route in the solution
► The total cost of the route $\rho$ is: $y_{\rho} = \sum_{k=1}^{r} c_{a_{k-1},a_k}$
► The penalty function for one route:

$$c_{\rho} = \begin{cases} y_{\rho} & \text{if } y_{\rho} \leq c_{\max} \\ \lambda(y_{\rho}, c_{\max}) & \text{otherwise} \end{cases}$$

► The objective function is:

$$\text{Minimize } c_{\text{total}} = \sum_{\text{all routes}} c_{\rho}$$

MIRHA approach

► The basic aspects of our proposed algorithm is the use of classical greedy heuristics combined with a bias randomization and a local search.
► Multi-start approach.
► Three main aspect:
  - Greedy classical heuristics
  - Bias random distribution combined with the greedy heuristic
  - Local search
MIRHA approach

- Get random solutions
  - Apply a Greedy Classical Heuristics randomized using a bias distribution:
    * The main idea of these heuristics is to select the next step from a list of available movements, usually according to a greedy criterion.
    * we consider non-uniform and nonsymmetric (biased) distributions, e.g.: the geometric distribution or the decreasing triangular distribution.
  - Local Search

procedure MIRHA(inputData, endConditions, prob.Dist., seed, heuristic)
- initializeRandomGenerator(seed);
- while endCondition[1] = false do
  * solution = getRandomSolution(inputData, heuristic, prob.Dist.);
  * solution = adaptiveLocalSearch(solution, endCondition[2]);
  * bestSolution = updateBestSolution(solution);
- end while;
- return bestSolution;
- end MIRHA
MIRHA approach

► Application to non-smooth/non-convex VRPs:
  ▪ Get random solutions
    * Apply the savings heuristic by Clarke and Wright, but a randomized version:
      – instead of having a single choice at every step, we will have multiple choices, each with a decreasing probability of being chosen.
  ▪ Local Search
    * The proposed method is an extension and improvement of the SR-GCWS-CS for the Capacitated Vehicle Routing Problem (CVRP).
    * Juan A., Faulin J., Jorba J., Riera D., Masip D., and Barrios B., 2010

MIRHA approach

► Characteristics of the MIRHA
  ▪ one important advantage of the proposed algorithm is its robustness and simplicity
    * MIRHA employs very few or no parameters, so there is no need to perform a complex fine-tuning process before using it.
  ▪ the MIRHA is simple to design and implement
    * for most of the combinatorial optimization problem there exist a classical greedy algorithm and a local search.
Computational results

► Classical CVRP benchmark instances from Christofides, Mingozzi and Toth
  • which feature the special constraints of the problem considered.
► The MIRHA algorithm has been implemented as a Java application.
► A standard personal computer, with an Intel R CoreTM2 Duo CPU processor at 2:4 GHz and a 2 GB RAM, was used to perform all tests.

Penalty function used:

\[ \lambda(y_p, c_{\text{max}}) = y_p + \min\{\theta(y_p, c_{\text{max}}), 20\} \]

where

\[ \theta(y_p, c_{\text{max}}) = 5 + 5000 \left( \frac{y_p - c_{\text{max}}}{y_p} \right)^4 \]
Computational results

► Comparison with

- **BKS-H**
- **CWS-H**
  * CWS heuristic Clark and Wright (1964) when considering hard constraints (CWS-H)
- **MIRHA-H**
- **GRASP-H**
  * Best solution obtained by GRASP using a restricted candidate list considering only $k$ percent of edges

Comparison among BKS-H, CWS-H, MIRHA-H and GRASP-H

- **Hard version**

<table>
<thead>
<tr>
<th>Instances</th>
<th>BKS-H</th>
<th>CWS-H</th>
<th>GAP</th>
<th>MIRHA-H</th>
<th>GAP</th>
<th>GRASP-H RCL=10%</th>
<th>GAP</th>
<th>GRASP-H RCL=50%</th>
<th>GAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrpnc6</td>
<td>555.43</td>
<td>618.39</td>
<td>11.34%</td>
<td>555.43</td>
<td>0.00%</td>
<td>581.76</td>
<td>4.74%</td>
<td>557.49</td>
<td>0.37%</td>
</tr>
<tr>
<td>vrpnc7</td>
<td>909.46</td>
<td>975.46</td>
<td>7.23%</td>
<td>915.4</td>
<td>0.63%</td>
<td>927.26</td>
<td>1.93%</td>
<td>975.46</td>
<td>7.23%</td>
</tr>
<tr>
<td>vrpnc8</td>
<td>865.94</td>
<td>973.94</td>
<td>12.47%</td>
<td>867.58</td>
<td>0.19%</td>
<td>899.52</td>
<td>3.88%</td>
<td>973.94</td>
<td>12.47%</td>
</tr>
<tr>
<td>vrpnc9</td>
<td>1162.55</td>
<td>1287.64</td>
<td>10.76%</td>
<td>1188.63</td>
<td>2.24%</td>
<td>1189.85</td>
<td>2.35%</td>
<td>1287.64</td>
<td>10.76%</td>
</tr>
<tr>
<td>vrpnc10</td>
<td>1395.85</td>
<td>1538.66</td>
<td>10.23%</td>
<td>1435.79</td>
<td>2.86%</td>
<td>1457.7</td>
<td>4.43%</td>
<td>1538.66</td>
<td>10.23%</td>
</tr>
<tr>
<td>vrpnc13</td>
<td>1541.14</td>
<td>1592.26</td>
<td>3.32%</td>
<td>1547.79</td>
<td>0.43%</td>
<td>1592.26</td>
<td>3.32%</td>
<td>1592.26</td>
<td>3.32%</td>
</tr>
<tr>
<td>vrpnc14</td>
<td>866.37</td>
<td>875.75</td>
<td>1.08%</td>
<td>866.37</td>
<td>0.00%</td>
<td>866.37</td>
<td>0.00%</td>
<td>875.75</td>
<td>1.08%</td>
</tr>
<tr>
<td>Averages</td>
<td>8.06%</td>
<td>0.91%</td>
<td>2.95%</td>
<td>6.49%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Averages 8.06% 0.91% 2.95% 6.49%
Computational results

- Comparison among BKS-H, CWS-S, MIRHA-S and GRASP-S
  - Soft version

<table>
<thead>
<tr>
<th>Instances</th>
<th>BKS-H</th>
<th>CW-S</th>
<th>GAP</th>
<th>MIRHA-S</th>
<th>GAP</th>
<th>GRASP-S RCL=10%</th>
<th>GAP</th>
<th>GRASP-S RCL=50%</th>
<th>GAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>vrpc6</td>
<td>555.43</td>
<td>629.88</td>
<td>13.40%</td>
<td>534.78</td>
<td>-3.72%</td>
<td>558.27</td>
<td>0.51%</td>
<td>534.8</td>
<td>-3.71%</td>
</tr>
<tr>
<td>vrpc7</td>
<td>909.68</td>
<td>975.89</td>
<td>7.28%</td>
<td>874.84</td>
<td>-3.83%</td>
<td>892.7</td>
<td>-1.87%</td>
<td>975.89</td>
<td>7.28%</td>
</tr>
<tr>
<td>vrpc8</td>
<td>865.94</td>
<td>941.17</td>
<td>8.69%</td>
<td>848.73</td>
<td>-1.99%</td>
<td>873.3</td>
<td>0.85%</td>
<td>941.17</td>
<td>8.69%</td>
</tr>
<tr>
<td>vrpc9</td>
<td>1162.55</td>
<td>1252.59</td>
<td>7.75%</td>
<td>1112.38</td>
<td>-4.32%</td>
<td>1134.34</td>
<td>-2.43%</td>
<td>1252.59</td>
<td>7.75%</td>
</tr>
<tr>
<td>vrpc10</td>
<td>1395.85</td>
<td>1475.57</td>
<td>5.71%</td>
<td>1389.32</td>
<td>-0.47%</td>
<td>1391.23</td>
<td>-0.33%</td>
<td>1475.57</td>
<td>5.71%</td>
</tr>
<tr>
<td>vrpc13</td>
<td>1541.54</td>
<td>1194.52</td>
<td>-22.49%</td>
<td>1139.22</td>
<td>-26.08%</td>
<td>1145.58</td>
<td>-25.67%</td>
<td>1194.52</td>
<td>-22.49%</td>
</tr>
<tr>
<td>vrpc14</td>
<td>866.37</td>
<td>868.68</td>
<td>0.27%</td>
<td>838.63</td>
<td>-3.20%</td>
<td>838.64</td>
<td>-3.20%</td>
<td>866.68</td>
<td>0.04%</td>
</tr>
<tr>
<td>Averages</td>
<td>2.94%</td>
<td>-6.23%</td>
<td>-4.59%</td>
<td>0.46%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* We can conclude the method is efficient and accurate.

Conclusions

- The multi-start biased randomization of classical heuristics with adaptive local search (MIRHA) algorithm is proposed as a method for solving non-smooth/non-convex vehicle routing problems.
- The key idea in our approach is to employ non-uniform and bias probability distributions such as the geometric to add a random biased behavior to classical heuristics, e.g. the savings method.
Conclusions

► Our methodology has similarities with several antecedent methods (GRASP, HBSS) but, at the same time, it maintains significant differences that have already been discussed.

► Computational results show the efficiency of our approach when dealing with VRP with non-smooth objective functions.

Further Research

► Application of the MIRHA approach to other non-smooth / non-convex problems as for example scheduling problems.

► Improve the MIRHA approach by studying the most adequate bias random distributions.