Hyper-Heuristics, Grammatical Evolution and the Capacitated Vehicle Routing Problem

Richard J. Marshall  
Victoria University of Wellington  
Wellington, New Zealand  
Richard.Marshall@msor.vuw.ac.nz

Mark Johnston  
Victoria University of Wellington  
Wellington, New Zealand  
Mark.Johnston@vuw.ac.nz

Mengjie Zhang  
Victoria University of Wellington  
Wellington, New Zealand  
Mengjie.Zhang@vuw.ac.nz

ABSTRACT
A common problem when applying heuristics is that they often perform well on some problem instances, but poorly on others. We develop a hyper-heuristic approach, using Grammatical Evolution (GE), to generate heuristics for the Vehicle Routing Problem (VRP). Through a series of experiments we develop an approach that leads to solutions of acceptable quality to Vehicle Routing Problem instances with only limited prior knowledge of the problem to be solved.

Categories and Subject Descriptors
I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—Heuristic methods

General Terms
Evolutionary Combinatorial Optimisation & Metaheuristics

Keywords
Routing & layout, Genetic Programming, Heuristics, Combinatorial Optimisation

1. INTRODUCTION
The VRP has wide ranging application in the transport and logistics industry. In the Capacitated Vehicle Routing Problem (CVRP) [7] a single depot holds a fleet of identical vehicles. A set of customers, each at a known location and with a known demand, are to be serviced. The objective is to service all customers while travelling the shortest possible total distance. Each customer must be serviced only once (split deliveries across multiple routes are not permitted), and each vehicle’s capacity must never be exceeded.

Design of heuristics for solving the VRP requires a trade-off between computation speed and achieving the best possible solution. Meta-heuristics have been developed over the last 40 years using different techniques which seek improvements to an initial solution by searching the adjacent and/or wider solution space. Many of these achieve good results but are often complex and time consuming to design and execute. More recent research has looked at hyper-heuristics which search a space of heuristics [5], as opposed to searching the space of solutions directly.

Here we use a hyper-heuristic approach to progressively build and search for improvements in parallel. This enables application of a search operation to a partially built solution. We follow the example of Burke et al. [1, 2] and generate new heuristics from the operations (or components) of existing heuristics.

Grammatical Evolution (GE) [6] belongs to a wider family of Grammar Guided Genetic Programming (GGGP) techniques [4] which emerged in the mid 1990s [8]. The benefits of GGGP over standard Genetic Programming include the ability of the grammar to restrict the search space and reduce the likelihood of semantically meaningless output.

The goal in this paper is to use a hyper-heuristic approach to evolve good heuristics that progressively construct and improve a solution until a complete solution to a CVRP instance is achieved in a reasonable computation time.

2. METHOD
The proposed hyper-heuristic approach is developed through a series of experiments which progressively expand the number and range of the operations able to be selected. A CVRP solution consists of a set of one or more vehicle routes, beginning and ending at the depot, with each route listing the customers in the order in which they are visited. The solution is complete if all customers are visited exactly once, and feasible if each vehicle’s capacity is never exceeded. Each operation performs an action on the current partial solution that either selects a customer and/or modifies one or more routes.

A heuristic consists of four distinct elements:
1. A strategy which creates an initial (partial) solution and defines how the final solution is to be developed.
2. A sequence of one or more operations (excluding a search operation) to construct a feasible solution.
3. A search operation to improve the current (partial) solution.
4. The number of times the whole sequence of operations (including the search operation) is repeated to deliver a final solution. We refer to each repetition of the sequence of operations as a cycle.

The search operation used is a deterministic variation of Iterated Local Search (ILS) [3]. Routes are chosen by firstly...
pairing neighbouring customers within a defined range and then sorting the pairs by distance between the customers (closest first). The pairs are then processed iteratively until the queue is empty.

We use GE to select the operations and their respective parameters. An example of the Backus Naur Form grammar used in our experiments is illustrated below. The only element of a heuristic that is not specified in the grammar is the number of cycles.

\[
\text{<strategy>::= <initialise>; <action> <search>;}
\text{<initialise>::= blank | cheapest | largest | farthest | nearest}
\text{<action>::= <build>; | <build> <action>}
\text{<build>::= <select>,<num>,<replace>}
\text{<search>::= 2opt | 3opt | ILSearch,<num>,<optType>}
\text{<select>::= cheapest | largest | farthest | nearest | remotest}
\text{<num>::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10}
\text{<replace>::= 0 | 1}
\text{<optType>::= 2 | 3}
\]

A typical example takes the form:

farthest; largest, 3, 0; cheapest, 6, 1; ... ILSearch, 8, 3.

Table 1: Results (abridged): Best 8 & worst 8 (of 40) compared to published best solution.

<table>
<thead>
<tr>
<th>CVRP</th>
<th>Off-line</th>
<th>On-line</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-n32-k5</td>
<td>787</td>
<td>787</td>
<td>788</td>
</tr>
<tr>
<td>A-n33-k5</td>
<td>668</td>
<td>662</td>
<td>663</td>
</tr>
<tr>
<td>A-n37-k5</td>
<td>672</td>
<td>672</td>
<td>673</td>
</tr>
<tr>
<td>A-n37-k6</td>
<td>958</td>
<td>951</td>
<td>952</td>
</tr>
<tr>
<td>A-n53-k7</td>
<td>1020</td>
<td>1012</td>
<td>1013</td>
</tr>
<tr>
<td>E-n23-k3</td>
<td>569</td>
<td>569</td>
<td>569</td>
</tr>
<tr>
<td>E-n33-k4</td>
<td>838</td>
<td>838</td>
<td>839</td>
</tr>
<tr>
<td>E-n76-k10</td>
<td>842</td>
<td>837</td>
<td>837</td>
</tr>
<tr>
<td>Worst:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A-n45-k6</td>
<td>954</td>
<td>950</td>
<td>945</td>
</tr>
<tr>
<td>A-n61-k9</td>
<td>1051</td>
<td>1048</td>
<td>1039</td>
</tr>
<tr>
<td>A-n62-k8</td>
<td>1303</td>
<td>1302</td>
<td>1294</td>
</tr>
<tr>
<td>A-n63-k9</td>
<td>1638</td>
<td>1634</td>
<td>1622</td>
</tr>
<tr>
<td>A-n63-k10</td>
<td>1323</td>
<td>1319</td>
<td>1314</td>
</tr>
<tr>
<td>A-n64-k9</td>
<td>1428</td>
<td>1416</td>
<td>1401</td>
</tr>
<tr>
<td>A-n65-k9</td>
<td>1192</td>
<td>1185</td>
<td>1182</td>
</tr>
<tr>
<td>A-n80-k10</td>
<td>1792</td>
<td>1770</td>
<td>1767</td>
</tr>
</tbody>
</table>

3. EXPERIMENT DESIGN AND RESULTS

Each of the experiments is run twenty times over each of 40 CVRP instances from http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/. The initial experiments use a starting population of 100 over 1,000 generations. The final experiment reduced this to a population of 40 over 10 generations without an appreciable loss of heuristic quality. The GE implementation uses integer genotypes with no wrapping allowed. Single point and sub-tree crossover, and intFlip and sub-tree mutation operators were trialled in different combinations.

The proposed hyper-heuristic approach is aimed at providing a minimal subset of operations that enable a good quality heuristic to be consistently developed. Our experiments trial both on-line and off-line learning. With off-line learning we use a different set of six CVRP instances for training and apply ten resulting heuristics on the test instances.

In our final experiment we introduce the Iterated Local Search operation described in Section 2 which is more closely aligned to a meta-heuristic approach. In common with other meta-heuristic search operations, the additional functionality comes at the cost of a fifteen-fold increase in computational time. Notwithstanding the large decrease in both population (100 to 40) and generations (1,000 to 10) when the search operation is enabled, the heuristics developed in the final experiment achieved the best found solutions from all our experiments for all 40 test instances. The best and worst solution distances from our experiments are given in Table 1. These are compared to the “best” solutions provided by the source web site.

The experiments show that it is possible to apply a hyper-heuristic approach to select operations which create a heuristic for a CVRP instance that delivers a high quality solution.

The decision to use GE has provided challenges. When using the sub-tree crossover and mutation operators we observe an early and rapid decline in the diversity of the population. This means the best solution that GE is likely to deliver in any given run is found in relatively few generations.

4. CONCLUSIONS

Results from applying the proposed hyper-heuristic method using GE indicate that good heuristics can be delivered that use operations that both construct and improve a solution to a CVRP instance. Further research will examine to what extent those improvements can be found in modifying the feedback provided to the search engine, streamlining the search operation, or using a different evolutionary search process.

5. REFERENCES


